AUXILIARY SUPERVISION FOR REGULARIZING DEEP LEARNING BASED IMAGE CLASSIFICATION

ZIPEI YAN

MPhil

The Hong Kong Polytechnic University
2023

The Hong Kong Polytechnic University

Department of Computing

Auxiliary Supervision for Regularizing Deep Learning Based Image Classification

Zipei Yan

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Philosophy

February 2023

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

Signature: _____

Name of Student: YAN Zipei

Abstract

Image classification is a fundamental task in visual recognition. Deep learning-based methods, i.e., Deep Neural Networks (DNNs), are state-of-the-art approach that achieves remarkable performance. Besides, DNNs pre-trained on image classification tasks with large-scale datasets show excellent transferability for solving downstream tasks, such as semantic segmentation, object detection, etc. Therefore, image classification becomes one of the fundamental but critical tasks in visual recognition. However, DNNs easily overfit and are hard to optimize, as they have billions or millions of parameters. To tackle this challenge, regularization techniques such as data augmentations and auxiliary learning are introduced to auxiliary supervise DNNs to achieve better generalization and robustness.

In this thesis, we first review existing regularization techniques in terms of data augmentation and auxiliary learning. Then we conduct two research works for regularizing DNNs on the classification task. More specifically, in the first work, we study the problem of computational color naming (CCN). We explore utilizing domain knowledge of the RGB Color Model as auxiliary supervision to regularize the model. Based on this, we expand CCN's application to data augmentation by designing a new data augmentation method named Partial Color Jittering(PCJ). PCJ performs the color jittering on a subset of pixels of the same image color,

which significantly increases images' diversity, thereby consistently improving image classification performance. In the second work, we study the problem in vision loss estimation. We first explore that vanilla models easily overfit and fall into trivial solutions in vision loss estimation. To tackle this challenge, we propose a novel method for vision loss estimation. In detail, we formulate VF estimation as an ordinal classification problem, following the ordinal properties of the studied data. Besides, we introduce an auxiliary task to assist the generalization of the model, where the auxiliary task explicitly regularizes the model. Finally, we conclude this thesis, discuss the open challenges and address future directions.

Publications Arising from the thesis

1. Zipei Yan, Linchuan Xu, Atsushi Suzuki, Jing Wang, Jiannong Cao, Jun Huang, “RGB Color Model Aware Computational Color Naming and Its Application to Data Augmentation”, in *IEEE International Conference on Big Data (Big Data)*, 2022.

2. Zipei Yan, Dong Liang, Linchuan Xu, Zhengji Liu, Jiahang Li, Jiannong Cao, Chea-su Kee, “Vision Loss Estimation using Fundus Photograph for High Myopia”, in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2023.

Acknowledgements

I would like to take this opportunity to express my gratitude to everyone who has helped and encouraged me throughout my master's study. Without your support and encouragement, I could not get through the difficulties during my study.

First of all, I would like to express my heartfelt gratitude to my supervisor, Dr. XU Linchuan, who has generously provided me with invaluable help and encouragement during my master's study. It is fortunate for me to be one of Dr. XU's students, and I have learned a variety of methodologies and skills from him. I have learned critical thinking, problem formulation, design solution, and academic writing. These knowledge and skills will consistently benefit my future studies.

Besides, I am grateful to Prof. CAO Jiannong, who co-supervises my master's study. And it is also an honor for me to attend the IMCL weekly meeting, where I meet many outstanding researchers and talented students. In addition, I'd like to thank all IMCL members for sharing their research works, which broadened my horizons.

In addition, I'm grateful to the committee members who participated in the oral defense of confirmation of my registration. They are Prof. CAO, Dr. XU, Dr. CHUNG Fu-Lai (Korris), and Dr. LIN Wanyu. I want to thank them for their constructive suggestions and generous encouragement. Besides, I'd like to express

my gratitude to the examiners of my oral examination, and they are Dr. CHUNG Fu Lai Korris, Prof. Arlindo Oliveira, Dr. CHEUNG Yim Lui Carol, and Dr. XU Linchuan.

Moreover, I'd like to thank my former roommates in the Student Halls of Residence (Hung Hom). They are Dr. WANG Shiqiang, Dr. Liu Junzhi, and Mr. PANG Kaicheng. I also want to thank Mr. SHI Qing and Mr. GONG Zheng, whom I got to know before I registered at PolyU and now Ph.D. students at PolyU. Besides, I'd like to express my gratitude to Mr. LIANG Dong, Mr. LIU Zhengji, and Mr. LI Jiahang, with whom I have collaborated on the research project. Also, I'd like to thank all PolyU staff.

Also, I'd like to express my gratitude to Miss TAN Jinyan, Mr. WANG Jingyu, and Mr. ZHAO Maoyu, with whom I spent the first new year in Hong Kong. It is a kind of predestination for us to study in Hong Kong and united together. We are all from the same hometown. Mr. WANG Jingyu and Miss TAN Jinyan are my high school classmates, and Mr. WANG Jingyu and Mr. ZHAO Maoyu are my PolyU schoolmates.

Furthermore, I would like to thank a particular person here, my girlfriend, Miss LI Qing, who has been my companion since my undergraduate days. I think both of us have experienced tough days since the COVID pandemic.

Finally, I would like to thank my parents and family. Without their support, I couldn't have made it this far.

Again, I'd like to thank those who have helped and encouraged me, but I can't specify their names at this moment.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Image classification is a fundamental task in visual recognition that aims to classify images into different categories. Before the era of deep learning, hand-crafted feature extraction is the dominant approach that manually extracts informative or descriptive features from images [1]. Then, extracted features are utilized to form a definition of each category. At inference/prediction time, a new image is classified into a category if its extracted features overlap significantly with an existing definition. The main difficulty in feature extraction is determining or choosing the extracted features from a given image [1]. As the number of categories increases, this difficulty becomes more troublesome.

With the rise of deep learning [2], deep neural networks (DNNs) achieve new state-of-the-art performance. DNNs are generally composed of the feature extractor and classifier. The feature extractor consists of deep combinations of operators (e.g., convolution, self-attention), activation functions (e.g., ReLU, GELU),

etc., for extracting the feature representations from images. After that, the classifier is to classify the feature representations into different categories by linear combination or non-linear transformation. As learning from a large amount of data with human-annotated labels, DNNs achieve remarkable performance, e.g., comparable accuracy to humans on ImageNet [3], [4]. Besides, DNNs trained on large-scale image classification datasets show excellent transferability for solving downstream tasks, such as semantic segmentation, object detection, etc.

Therefore, DNNs with high performance on image classification tasks are always desired. Thereafter, how to boost the performance of DNNs on image classification tasks becomes a fundamental problem. To tackle this challenge, regularization [5], [6] is introduced to make the DNNs generalize better. Apart from the standard optimization procedure for training DNNs, extensive research works focus on different regularization techniques that aim to make the DNNs generalize better, such as data augmentation, auxiliary tasks, etc. These regularization techniques involve auxiliary supervision for regularizing DNNs to generalize better. For example, MAXL [7] introduces auxiliary tasks for supervising DNNs to better generalization. Besides, Mixup [8] augments the training data and introduces an extra supervision signal as it linearly interpolates both training data and related labels. In addition, the success of larger DNNs is strongly linked to regularization techniques. Because the larger DNNs are typically millions or billions of parameters and are difficult to optimize, whereas these techniques significantly regularize DNNs during training. Besides, even the 'classic' DNNs can be revived by these regularization techniques, e.g., ResNet-50 strikes back with improved training procedure [9], thereby further demonstrating the effectiveness and necessity of regularization.

1.2 Research Challenges

The main challenge in improving performance through regularization as auxiliary supervision lies in how to design a regularization.

Data augmentation is one efficient method for regularizing DNNs. Designing a data augmentation is mainly based on prior knowledge. For instance, geometric transformations, such as cropping, rotating, and flipping, are performed to encode invariant priors so that trained DNNs can generalize better with these invariant priors. However, these priors are not always helpful in some cases that conflict with domain knowledge. For example, in the fine-grained classification where DNNs are trained to classifier hard-to-distinguish objects, such as flowers [10] and birds [11], data augmentations such as color jittering would hurt the model's generalization [12]–[14]. The main reason is that the color is sufficiently correlated to its categories, e.g., two flowers with the same shape but with different colors are classified into two categories, and after color jittering, the primary distinguishable characteristic is lost. Besides, data augmentation introduces a new bias to the model. For instance, ImageNet pre-trained models are biased to the image's texture [15], such that the model is easily attacked by the textures. Then, a natural way to improve the model's robustness and generalization is to de-bias it to textures by introducing shape bias from data augmentation [15].

Auxiliary learning is another effective regularization, which improves the generalization of primary tasks by introducing auxiliary tasks. It is usually assumed that the auxiliary task should be related to the primary task in some way, and thus solving it can be helpful for the primary task [16]. The main challenge in auxiliary learning is to find the auxiliary tasks that are sufficiently related to the primary

task. Utilizing the domain or prior knowledge to manually find auxiliary tasks is the most commonly adopted approach when given the domain knowledge of the main task. However, it is costly to manually examine whether each auxiliary task is helpful or not. Besides, how to properly utilize the auxiliary task is another challenge. The auxiliary task is not always helpful for the primary task, because sometimes a negative transfer exists from the auxiliary task to the primary task [17]–[19].

1.3 Thesis Contributions

The contributions of this thesis mainly lie in designing novel regularization methods to auxiliary supervise DNNs in order to achieve better generalization.

Firstly, we study the problem in computational color naming (CCN). Computational color naming (CCN) aims to learn a mapping from pixels into semantic color names. Existing research on CCN mainly studies pixels collected in laboratory settings or images collected from the web. However, laboratory pixels are in a limited data size such that the learned mapping may not generalize well on unseen pixels, and the mapping discovered from images is usually data-specific. Therefore, we aim to learn a universal mapping by studying pixels collected from the web. To achieve this objective, we formulate a novel classification problem that incorporates both the pixels and the RGB color model. The RGB color model is beneficial for learning the mapping because it characterizes the production of colors, e.g., adding red and green produces yellow. However, the characterization is rather qualitative. To solve this problem, we propose ColorMLP, a multilayer perceptron (MLP) embedded with graph attention networks (GATs). Here, the

GATs are designed to capture color relations that we construct by referring to the RGB color model. In this way, the parameters of the MLP can be regularized to comply with the RGB model. We conduct extensive experiments to demonstrate the superiority of ColorMLP to alternative methods. Besides, we design a novel data augmentation method named partial color jitter (PCJ) to expand the application of CCN. PCJ performs color jitter (CJ) on a subset of pixels of the same image color. In this way, PCJ partially changes the color properties of images, thereby significantly increasing images' diversity. We conduct extensive experiments on CIFAR-10/100 and ImageNet datasets, showing that PCJ can consistently improve classification performance.

Second, we study the problem in vision loss estimation. Visual field (VF) sensitivity is a commonly used metric to quantify vision loss; it is a crucial criterion for diagnosing high myopia (HM) complications. However, measuring VF is prohibitively time-consuming and subjective as it highly depends on patient compliance. Consequently, utilizing machine learning models to estimate VF becomes a feasible alternative. Fundus photographs have become the preferred modality for studying HM, due to their convenience of acquisition and incorporation of structural information. Conversely, estimating VF with vanilla regression using fundus photographs falls into trivial solutions. To tackle this challenge, we propose a novel method for VF estimation. In detail, we formulate VF estimation as an ordinal classification problem, where each VF point is interpreted as an ordinal variable rather than a continuous one, given that any VFS point is a discrete integer with a relative ordering. Besides, we introduce an auxiliary task for myopic maculopathy classification to assist the generalization of VFS estimation. Myopic maculopathy (MM) is strongly associated with vision loss, and its symptoms can

be observed from the fundus photographs directly; therefore, the model will be explicitly regularized by the auxiliary information if utilized properly. Not only does our method outperform vanilla baseline by 15% on a clinic-collected real-world dataset, but it can also be utilized to detect potential vision loss for HM cases in large-scale preliminary selection.

1.4 Thesis Organization

The rest of this thesis includes a literature review on regularization and auxiliary supervision, two research works for designing novel regularization methods to auxiliary supervise DNNs in practical applications, conclusions, and a discussion of open challenges and future directions.

More specifically, the rest chapters of the thesis are organized as follows:

- In Chapter 2, we first review the related literature on regularization and auxiliary supervision. Besides, we review data augmentation and auxiliary learning in terms of regularization and auxiliary supervision.

- In Chapter 3, we present our first work on computational color naming (CCN) and further expand CCN's application to data augmentation. We first review the problem in CCN; then, we propose a novel model named ColorMLP for CCN by additionally utilizing the RGB Color Model as regularization. Besides, we expand CCN's application to data augmentation by designing a color jittering-based data augmentation method, namely Partial Color Jitter, which performs CJ on a subset of pixels belonging to the same color of an image. In this way, PCJ partially changes the color properties

of images, thereby significantly increasing images' diversity. We conduct experiments to show that PCJ has a remarkable regularization effect on the image classification tasks.

- In Chapter 4, we present our second work on vision loss estimation. We first review the problem in vision loss estimation and find out that existing vanilla baselines produce trivial solutions and thus fail to estimate vision loss accurately. To tackle this challenge, we propose a novel method based on the characteristics of Visual field (VF) sensitivity data. Besides, we introduce an auxiliary task for myopic maculopathy classification to assist the generalization of vision loss estimation. Finally, we conduct experiments to evaluate our method on a clinic-collected real-world dataset.

- In Chapter 5, we conclude the thesis and discuss the open challenges and the directions of future works. The open challenges are mainly about data augmentation and auxiliary learning. For data augmentation, theoretical research for analysis data augmentation is an open challenge. Besides, analyzing the effect of data augmentation on the transferability of pre-trained DNNs is an open challenge. And how to efficiently design a general data augmentation that does not rely on prior knowledge is also an open challenge. For auxiliary learning, finding sufficient related auxiliary tasks is a basic challenge. Besides, eliminating the negative transfer from auxiliary tasks is another open challenge. Based on the above open challenges, future directions are as follows. From the representation learning perspective to analyze data augmentation is one interesting future direction. In addition, based on the representation learning perspective, latent space data

augmentation is a promising future direction, which is modality agnostic and more general, as it decouples from input modality and only generates or augments deep features in the latent space. In terms of auxiliary learning, self-supervised auxiliary learning is a promising future direction to explore, where the auxiliary labels are obtained from the data itself, then the manual procedure is no longer needed, hence it will be more efficient and more general.

Chapter 2

Literature Review

In this chapter, we first review the definition of regularization and auxiliary supervision, then discuss two representative techniques, including data augmentation and auxiliary learning. Besides, we review the existing works on computational color naming and visual field sensitivity estimation.

2.1 Regularization and Auxiliary Supervision

The term "regularization" is first used as a penalty term in the loss function [20]. Goodfellow *et al.* have broadened its meaning as: "any modification we make to a learning algorithm that is intended to reduce its test error but not its training error" [5]. Then, a slightly restrictive definition is "Regularization is any supplementary technique that aims at making the model generalize better, *i.e.*, produce better results on the test set" [6]. In conclusion, we define "regularization" as the technique that improves the model's generalization.

To formalize auxiliary supervision, let's consider a supervised learning setting.

We are given a training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=0}^{n}$ where $\boldsymbol{x}_i$ denotes input data, y_i denotes the supervision signal, a deep neural network $f(\cdot;\theta)$ where θ denotes its parameters. The overall objective is to find a target θ^* by minimizing a designed loss function $\mathcal{L}$ on $\mathcal{D}$, which is generally formulated as follows:

$$\theta^* = \arg\min_{\theta} \frac{1}{|\mathcal{D}|} \sum \mathcal{L}_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}}(f(\boldsymbol{x}_i;\theta), y_i) + \mathcal{R}(\theta) \tag{2.1}$$

According to the above objective, the following components are naturally connected to θ^*, thus determining the final performance:

- $\mathcal{D}$: the training data.
- $f(\cdot;\theta)$: the model.
- $\mathcal{L}(\cdot)$: the loss function.
- $\mathcal{R}(\cdot)$: the penalty term/auxiliary loss function.

Therefore, "any supplementary techniques" on training data, model, loss function, and auxiliary loss function are so-called regularizations. And these regularizations act as auxiliary supervision, as they additionally regularize the model to better generalization.

2.1.1 Data Augmentation

Data augmentation is a widely applied regularization for relieving the overfitting problem by increasing the dataset size and diversity [21]. Different from the taxonomy in [21], we mainly review data augmentation in two aspects according to whether it performs augmentation in the input or latent space. Specifically, data

augmentations in input space refer to these directly operating on the raw images. Besides, data augmentations in latent space generate augmented/transformed features in the deep latent space.

2.1.1.1 Data Augmentation in Input Space

Basic manipulations, including geometric and color transformations, are applied to the input space of the image, which changes the geometric and color properties of an image. These manipulations are directly applied to images, making them simple to implement. For geometric transformations, such as cropping, rotating, and flipping are performed to encode in-variance priors, which have been widely adopted as standard augmentation for training DNNs, especially for convolutional neural networks (CNNs). Besides, color transformations such as hue jittering, etc, are also frequently utilized. These basic transformations are usually applied sequentially during training.

After that, finding effective combinations of basic manipulations becomes a research problem. Different from manually finding combinations, AutoAugment [22] is proposed to automatically search the best combinations(/policies) according to the highest validation accuracy with Reinforcement Learning. AutoAugment has achieved amazing performance on both large-scale and tiny datasets. However, such a search algorithm typically takes too much time, although it can be run in parallel. Fast AutoAugment [23] proposes to find effective combinations by density matching, which reduces the search space and thus saves time. Besides, Population Based Augmentation (PBA) [24] generates dynamic augmentation policies instead of fixed ones. In addition, TrivialAugment [25] proposes a trivial yet efficient method, which uniformly selects one augmentation from a

given polices during the training. TrivialAugment achieves comparable performance compared to AutoAugment.

Erasing-based data augmentations also show remarkable performance. These augmentations typically erase one or some sub-regions of raw images. For example, Cutout [26] randomly erases a square sub-region of the raw image with constant values, which regularizes the model and improves its robustness. Besides, RandomErasing [27] first randomly chooses a rectangle region, then replaces this region with random values. RandomErasing has shown better performance than Cutout. In addition, GirdMask [28] generates a gird-like squared mask to mask out the raw image, which is based on the deletion of regions of the input image.

In addition, Mixup [8] style-like data augmentations perform data augment by mixing two input images. Specifically, Mixup [8] proposes to combine two images as well as their labels in a convex manner. Mixup not only regularizes DNN to better performance but also increases its robustness to adversarial examples. Instead of mixing the whole images, CutMix [29] proposes to cut and paste patches, thereby efficiently using training pixels and retaining the regularization effect of regional dropout. Unlike mixing different training samples, AugMix [30] proposes to mix three augmented samples from a single image in a convex combination. Besides, AugMix additionally utilizes Jensen-Shannon divergence consistency loss to enforce smoother neural network responses. AugMix improves DNNs' robustness and helps them withstand unforeseen corruption.

2.1.1.2 Data Augmentation in Latent Space

Unlike data augmentations in input space, data augmentations in latent space do not explicitly manipulate the raw image, as they typically operate on deep features

in latent space. Besides, these augmentations can generate different semantic samples, such as changing the view angle, changing the background color, etc., which usually cannot achieve by basic manipulations. For example, ISDA [31] achieves implicit semantic data augmentation by transforming deep features along semantic directions in latent space. Besides, [32] proposes a unified viewpoint between data augmentation and loss variations incurred by logit perturbation. In short, ISDA is formulated as a specific logit perturbation. Moreover, logit perturbation achieves better when facing imbalanced data compare to ISDA. Furthermore, generative adversarial networks (GANs) [33] are also introduced for generating more data. For example, [34] utilizes GANs to synthesize more training samples to improve liver lesion classification. GANs-based data augmentations can be formulated as special data augmentation in latent space because these methods first generate semantic features in latent space and then transform them back to the input image space. Moreover, Manifold Mixup [35] generalizes Mixup to the latent space, which mixes deep features in latent space and their corresponding labels. MODALS [36] propose an automated latent data augmentation method by searching the best data augmentation policies with PBA in latent space. MODALS achieves comparable performance to Mixup on different modalities data.

2.1.2 Auxiliary Learning

Auxiliary learning is a special type of multi-task learning in which the auxiliary tasks are introduced only to help generalize the primary task [7], [16]. Specifically, auxiliary learning only pays attention to the performance of the primary tasks, whereas multi-task learning aims to achieve comparable performance among all

tasks [16].

Auxiliary learning has succeeded in many applications, including computer vision, natural language processing, and speech recognition. For example, ROCK [37] utilizes auxiliary tasks for predicting scene labels and evaluating depth and surface orientation at a pixel level. Besides, [38] proposes to predict words based on their neighborhood as an auxiliary task, such that the model can learn efficient word representations. In addition, [39] utilizes phoneme recognition at intermediate low-level representations as an auxiliary task to improve conversational speech recognition performance. Besides, Auxiliary learning has been applied in Reinforcement Learning. For instance, [40] proposes two auxiliary tasks, including pixel changes and network features, to promote faster training, more robust learning, and ultimately higher performance. Besides, [41] proposes to use different feature spaces for computing prediction errors as auxiliary tasks to improve the model's generalization in Curiosity-Driven Learning.

The main challenge in auxiliary learning is to find the related auxiliary tasks. However, finding an auxiliary task is largely based on the assumption that the auxiliary task should be related to the primary task in some way; thus, solving it can be helpful for the primary task [16]. Unlike utilizing domain or prior knowledge to find auxiliary tasks manually, MAXL [7] proposes a self-supervisor label generator for generating auxiliary labels to assist the generalization of the primary task. Besides, the auxiliary task is not always helpful for the primary task, because sometimes there exists a negative transfer from the auxiliary task to the primary task. Therefore, quantifying whether the auxiliary task is helpful for the primary task becomes a crucial problem. [17] proposes that the cosine similarity between gradients from the auxiliary task and primary task can provide a signal to detect

when an auxiliary task is helpful to the primary task. Based on this, [17] demonstrates the negative transfer when such cosine similarity becomes negative. Then, a natural strategy for blocking the negative transfer is to mitigate harmful auxiliary gradients [17]. In detail, [17] proposes a weighted and unweighted strategy for mitigating harmful auxiliary gradients. Besides, [18] proposes to project the auxiliary gradients to the primary gradients, then remove the harmful ones. In addition, [19] proposes to decompose auxiliary gradients into directions that help, damage, or leave the primary task loss unchanged. Based on this decomposition, an efficient algorithm is proposed to re-weight the auxiliary gradients differently depending on their impact on the problem of interest.

2.2 Existing Works on Computational Color Naming

Computational color naming (CCN) aims to learn a mapping from pixels into semantic color names. CCN has wide applications. First, it can assist color-deficiency people in recognizing colors in the digital world. Second, CCN can help improve the performance of many visual recognition tasks such as image retrieval [42], object detection [43], visual tracking [44], and texture recognition [45]. According to the studied data, existing works on CCN are mainly two types.

The first type studies pixel-color pairs collected under laboratory settings [46]–[50]. In general, they ask several observers with no color deficiencies to distribute a total score of 10 points among the possible color names according to the certainty they had about each pixel belonging to the different categories. Therefore, only a

small number (e.g., hundreds) of pixel-color pairs is available, which may result in a poor generalization of the discovered mapping to unseen pixels.

The second type studies images collected from the web [51]–[53]. They typically collect color images from the Google image search engine by utilizing the input search key and its corresponding images, e.g., red rose. Such a process is more convenient than the first type in laboratory settings. However, collected images contain the target color and other irrelevant ones, which are not described by the target color name. Therefore, learning models based on these image-color pairs may lead to poor generalization, as there exist unwanted spurious correlations among image-color pairs.

2.3 Existing Works on Visual Field Sensitivity Estimation

Visual field (VF) sensitivity is a commonly used metric to quantify vision loss which can be measured by the visual field test. The visual field test is prohibitively time-consuming and subjective due to its high dependence on patient compliance [54]. Consequently, utilizing DNNs to estimate VF becomes a feasible alternative, because DNNs are capable of making fast predictions.

Existing works on VF estimation mainly utilize pre-trained DNNs to learn mappings from eye-related modalities to VF. These modalities contain potential vision loss information, such as retinal thickness and fundus photographs. Retinal thickness reflects the eye's functionality, typically degraded by vision loss diseases such as glaucoma. Besides, fundus photographs capture structural information

such as retinal vascular, and myopic macular, which can be utilized to diagnose myopic maculopathy, a myopia-related disease causing irreversible vision loss.

There are mainly two types of existing works according to their studied modality. The first one estimate VF only for the glaucomatous population by using different combinations of retinal thicknesses [54], [55]. The second one estimates quantitative measurements (e.g., MD value) in glaucoma by utilizing various types of fundus photographs [56], [57]. Notably, all these existing works are limited only to glaucoma.

Chapter 3

RGB Color Model Aware Computational Color Naming and Its Application to Data Augmentation

Computational color naming (CCN) aims to learn a mapping from pixels into semantic color names, e.g., red, green and blue. CCN has wide applications including color vision deficiency assistance and color image retrieval. Existing research on CCN mainly studies pixels collected under laboratory settings or studies images collected from the web. However, laboratory pixels are very limited such that the learned mapping may not generalize well on unseen pixels, and the mapping discovered from images is usually data-specific. In this work, we aim to learn a universal mapping by studying pixels collected from the web. To this end, we formulate a novel classification problem that incorporates both the pixels and the

RGB color model. The RGB color model is beneficial for learning the mapping because it characterizes the production of colors, e.g., the addition of red and green produces yellow. However, the characterization is rather qualitative. To solve this problem, we propose ColorMLP, which is a multilayer perceptron (MLP) embedded with graph attention networks (GATs). Here, the GATs are designed to capture color relations that we construct by referring to the RGB color model. In this way, the parameters of the MLP can be regularized to comply with the RGB model. We conduct comprehensive experiments to demonstrate the superiority of ColorMLP to alternative methods.

To expand the application of CCN, we design a novel data augmentation method named partial color jitter (PCJ), which performs color jitter (CJ) on a subset of pixels belonging to the same color of an image. In this way, PCJ partially changes the color properties of images, thereby significantly increasing images' diversity. We conduct extensive experiments on CIFAR-10/100 and ImageNet datasets, showing that PCJ can consistently improve the classification performance.

3.1 Introduction

Computational color naming (CCN) aims to learn a mapping from pixels into semantic color names. According to linguistics studies [58], color names vary in different languages, but most languages share 11 basic color names, i.e., black, white, red, green, yellow, blue, brown, orange, pink, purple, and gray. CCN has wide applications. First, it can assist color-deficiency people in recognizing colors in the digital world. Studies have discovered that the prevalence of color deficiency in European Caucasians is about 8% in men and about 0.4% in women

and between 4% and 6.5% in men of Chinese and Japanese ethnicity [59]. Such a prevalence of color deficiency worldwide can lead to great demands for color-deficiency assistance. Second, CCN can help improve the performance of many visual recognition tasks such as image retrieval [42], object detection [43], visual tracking [44], and texture recognition [45].

There are mainly two types of existing approaches to CCN according to studied data. The first type studies pixel-color pairs collected under laboratory settings [46]–[50]. In the laboratory, several observers with no color deficiencies were asked to distribute a total score of 10 points among the 11 possible color names according to the certainty they had about each pixel belonging to the different categories. It is worth mentioning that there are several Apps for color-deficiency assistance, e.g., ColorBlindPal and WhatColor. They also have a set of pixel-color pairs whose source is unclear to us. The second type studies images collected from the web [51]–[53]. One way to collect such images can be as follows: Google image search uses the image filename and surrounding web page text to retrieve the images [51].

However, both the two types of existing approaches have limitations. For the pixel-based approach, it is expensive to collect pixel-color pairs under laboratory settings. Therefore, only a small number (e.g., hundreds) of pixel-color pairs is available, which may result in a poor generalization of the discovered mapping to unseen pixels. For those color-deficiency assistance Apps, to our best knowledge, they usually find the nearest match to any given pixel. In other words, they employ the nearest neighbor approach. The performance of the nearest neighbor approach highly depends on their set of pixel-color pairs. For the image-based approach, it is more convenient to collect data from the web in which a color name is usually

given to the object of interest of an image (e.g., a red car). However, the object boundary is not given. Moreover, there can be a considerable number of pixels that are not parts of the object and that are not described by the color name. Therefore, the discovered mapping is usually data-specific.

In this work, we study pixel-color pairs collected from the web with the objective of learning a universal mapping. In particular, we collect a set of pixel-color pairs from multiple sources including some color standards and a survey. The objective is to learn a mapping for all the pixels in the RGB color space, i.e., the RGB cube as illustrated in Fig. 3.1. However, the set is still limited compared to all the pixels in the RGB cube that has as many as $256 \times 256 \times 256$ pixels. Therefore, instead of purely learning from the limited data, we propose to further incorporate the characteristics of the RGB color model. Based on the human perception of colors, the RGB color model [60] is an additive color model and characterizes how colors are produced, e.g., the addition of red and green produces yellow. However, it is challenging to inform a machine learning model of the RGB color model's characteristics since a machine learning model deals with quantitative computations while the characteristics are rather qualitative.

To address the challenge, we propose ColorMLP, which is a multilayer perceptron (MLP) embedded with graph attention networks (GATs) [61]. In particular, we construct graphs with colors as vertices by referring to the RGB color model. The GATs are designed to turn the graphs into the representations of colors, which are then used as parameters in the MLP. In this way, the parameters of the MLP are regularized to comply with the RGB model.

We further expand the application of CCN to data augmentation. Data augmentation plays a crucial role in regularizing deep neural networks. Color jitter

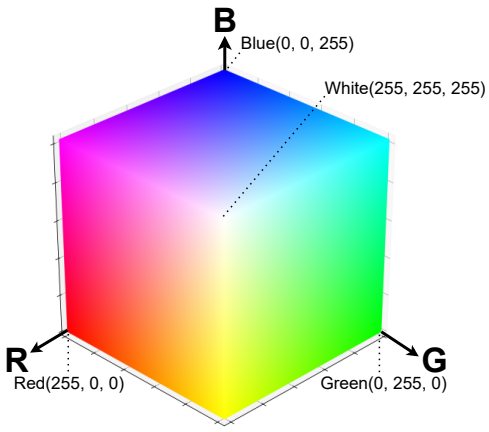


Figure 3.1: The illustration of the RGB cube. The bold-faced R, G, and B represent the red channel, the green channel, and the blue channel, respectively. Any point in the cube, e.g., (255, 255, 255), is a pixel.

(CJ) is the most commonly used color-involved augmentation methods. Specifically, CJ randomly changes the brightness, contrast and saturation of an image, thereby increasing the diversity of images. However, some existing studies [12], [13] have shown that CJ may degrade the performance of image classification. One potential reason is that CJ changes the color patterns of an image arbitrarily, which may introduce unrealistic color patterns of objects. To reduce the risk of unrealistic color patterns, we propose a new method named partial color jitter (PCJ), which only performs CJ on a subset of pixels with the same color of an image. In this way, PCJ can also improve the diversity brought by the color changes.

Our contributions are summarized as follows:

- We consolidate a dataset of pixel-color pairs for computational color naming. The dataset consists of pixels from different sources [1] including some color standards and a survey conducted on the web.

[1] All the sources grant a free license. URLs are included in section 3.6.

- We formulate CCN as a novel classification problem where pixel-color pairs and the RGB color model are given, and propose ColorMLP to solve the problem. We provide both qualitative and quantitative evaluations to demonstrate that ColorMLP significantly performs better than alternative classifiers that can only learn from the pixel-color pairs.

- We divide the entire RGB data space into 11 color regions corresponding to the 11 color names. The 11 color regions can be utilized to develop color-deficiency assistance applications.

- We expand the application of CCN to data augmentation by designing PCJ for image classification. We conduct extensive experiments on CIFAR-10/100 and ImageNet to demonstrate that PCJ can obtain the state-of-the-art performance.

The rest of this chapter is organized as follows. Section 3.2 introduces the related work. Section 3.3 presents the studied problem. Section 3.4 presents ColorMLP. Section 3.5 describes PCJ. Section 3.6 and 3.7 provide empirical evaluations of ColorMLP and PCJ, respectively. Section 3.8 concludes this work and introduces future work.

3.2 Literature Review

We discuss three types of related work w.r.t. to three topics, respectively, which are CCN, the integration of graph neural network (GNN) into MLP, and data augmentation.

In terms of CCN, there are mainly two types of related work according to the format of the data studied. The first type [46]–[50] studies pixel-color pairs. We also study pixel-color pairs but of a different kind. In particular, the related work studies pixel-color pairs collected under laboratory settings while we study pixel-color pairs collected from the web. Moreover, the color information for each pixel in the related work contains a membership value to each of the 11 colors such that related work performs fuzzy modeling of the data. By contrast, the data in our study specify a single color category for each pixel, and therefore we perform the conventional classification. The second type [51]–[53], [62] studies image-color pairs. This type focuses more on specific image applications such that the learned pixel-to-color mapping is data set specific. By contrast, we aim to learn a universal mapping. It is worth mentioning that CCN can be used in color-deficiency assistance, which can help the color blindness recognize colors in the digital world. There are some color-deficiency assistance Apps, e.g., ColorBlindPal and WhatColor. To our best knowledge, they also have a set of pixel-color pairs, and utilize the nearest neighbor approach. We are different from them in both the dataset and the approach.

The integration of GNN into MLP has been studied in multi-label image classification [63], [64]. In particular, GNNs are utilized to learn representations of class labels, and then the representations are used as the parameters of the fully connected layers. The main difference from us is that they deal with multi-label classification while we deal with multi-class classification.

Data augmentation is a technique for regularizing deep neural networks [21]. Geometric transformations such as cropping and flipping are frequently methods. Besides, color space transformations, particularly CJ, can be applied to colorful

images. In this work, we propose PCJ that performs CJ on parts of an image. It is worth mentioning that there are several studies related to CJ. Instead of manually selecting augmentation methods, studies [22], [23] design algorithms to automatically search over a set of augmentation methods including CJ for an improved augmentation. Note that CJ may not consistently bring benefits as suggested by some studies [12], [13]. We show that PCJ can consistently bring benefits. Therefore, PCJ may replace CJ in the search base. Another study [31] realizes color transformations through an optimization approach. In this way, it does not perform color transformations explicitly like PCJ.

3.3 The Studied Problem

We are given a set of pixel-color pairs $\{(\boldsymbol{p}_i, c_i)\}_{i=1}^{N}$, where $\boldsymbol{p}_i \in \mathbb{R}^3$ is the representation of a pixel denoted in the RGB format, c_i is the color name of the pixel, and N is the number of pixels. The RGB format denotes a pixel by a three-dimensional representation where the three dimensions correspond to three channels, i.e., Red channel, Green channel and Blue channel, respectively. We study the 11 basic color names shared by most languages in this work. Our objective is to learn a mapping from $\boldsymbol{p}_i$ into c_i by utilizing both $\{(\boldsymbol{p}_i, c_i)\}_{i=1}^{N}$ and the RGB color model. More information about the RGB color model will be introduced in the following section. The novelty of the problem is additionally regarding the characteristics of the RGB model as input, and the challenge mainly lies in how to turn the qualitative characteristics into quantitative computations.

3.4 ColorMLP

3.4.1 Design Rationale

To address the challenge, we learn that the RGB color model is an additive color model [60] and is based on three primary colors. Hereafter, Red, Green and Blue denote the three primary colors while red, green and blue denote the colors in the 11 basic colors. Each basic color is produced by the addition of some primary colors, e.g., the addition of Red and Green produces yellow. We figure out that two basic colors may be related according to whether a particular primary color is added for their productions or not. We thus propose to construct a graph with the 11 basic colors as vertices and color relations as edges. Afterwards, we may conduct computations on the graph. In particular, we employ GAT [61], a popular graph neural network, to deal with the graph.

What is left to be addressed is how to integrate the graph computation into the pixel-to-color mapping. The mapping can be typically discovered by fitting a classifier. We propose to design an MLP for this purpose because an MLP with sufficient capacity can approximate any arbitrary mapping function [65]. Moreover, we find it flexible to integrate GAT into MLP. As a result, we design a model named ColorMLP. Later on, we first give an overview of ColorMLP and then present the design details.

3.4.2 Overview

We present Fig. 3.2 to illustrate the proposed ColorMLP, which is drawn by following the notations of PyTorch. ColorMLP is a specially designed MLP with

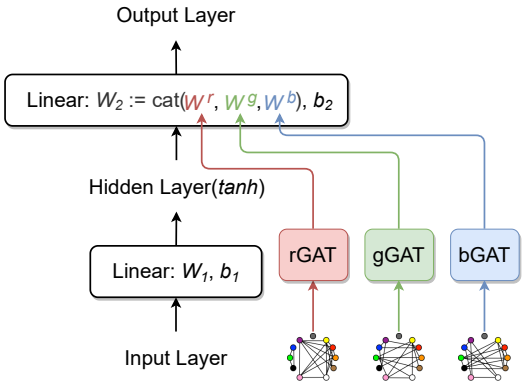


Figure 3.2: The illustration of ColorMLP. Following the convention of PyTorch, Linear represents the connection between two layers and consists of a weight matrix $\boldsymbol{W}_1$ and bias vector $\boldsymbol{b}_1$. tanh is the activation function utilized in the hidden layer. rGAT, gGAT and bGAT are three GATs we design to take three respective color graphs as input and produce three sets of color representations concatenated as the weight matrix $\boldsymbol{W}_2$ through cat($\boldsymbol{W}^r$, $\boldsymbol{W}^g$, $\boldsymbol{W}^b$).

one hidden layer. The difference of ColorMLP from the conventional MLP is just in the weight matrix of the connection between the hidden layer and the output layer. In particular, the weight matrix of ColorMLP is the concatenation of the outputs of three GATs. Each GAT takes as input a distinct graph of colors, and produces the matrix that consists of the continuous representations of the colors. The dimensions of the matrix are specified such that the concatenation of the three matrices is compatible with the design of the hidden layer. The three graphs share the same set of vertices (i.e., the 11 basic colors as vertices) but have different sets of edges. We define the edges according to the RGB color model. As a result, ColorMLP is an MLP embedded with domain knowledge of the RGB model, thereby regularizing the weight matrix to reduce its overfitting on data. In the following subsections, we present how the graphs are constructed, how the outputs of the GATs are integrated into the MLP, and how ColorMLP is optimized, respectively.

Table 3.1: Color and its related colors in terms of the RGB channels.

Color	Related Colors		
	Red Channel	Green Channel	Blue Channel
red	yellow, brown, orange, pink, purple, white	blue, purple, black	green, yellow, brown, orange, black
green	blue, black	yellow, brown, orange, pink, white	red, yellow, brown, orange, black
blue	green, black	red, purple, black	pink, purple, white
yellow	red, brown, orange, pink, purple, white	green, brown, orange, pink, white	red, green, brown, orange, black
brown	red, yellow, purple	green, yellow	red, green, yellow, orange, black
orange	red, yellow, pink, purple, white	green, yellow, pink	red, green, yellow, brown, black
pink	red, yellow, orange, purple, white	green, yellow, orange, white	blue, purple, white
purple	red, yellow, brown, orange, pink, white	red, blue, black	blue, pink, white
white	red, yellow, orange, pink, purple	green, yellow, pink	blue, pink, purple
gray	–	–	–
black	green, blue	red, blue, purple	red, green, yellow, brown, orange

3.4.3 Graph Construction

We first briefly introduce the RGB color model and then present our method for constructing the color graphs.

3.4.3.1 Introduction to the RGB color model

The RGB model is an additive color model with three primary colors [60]. It specifies that each primary color corresponds to a channel of a pixel and that each channel has an integral intensity ranging from 0 to 255 inclusively. When the Red channel has the strongest intensity, a red pixel is produced, e.g., an intensity of 200 in the Red channel and a zero intensity in both the Green channel the Blue channel produce a red pixel (200, 0, 0). If the same intensity in the Green channel is added, a yellow pixel (200, 200, 0) is produced. Pixels in other colors have different addition rules, which are omitted for space consideration. These pieces information is sufficient for us to establish the color relations.

3.4.3.2 Color relations

According to the RGB color model, two colors may share similar intensities in some channels, e.g., red and yellow share similar intensities in the Red channel. We thus propose to establish the color relations according to the shared channel-wise intensity. In particular, two colors are considered to be related in terms of a channel if the two colors share similar intensities in the channel. However, it is non-trivial to determine how similar the intensity should be.

To address this challenge, we propose a two-fold criterion. ***First, a channel representing a primary color of interest should be added to produce both the colors***. This is based on the additive property of the RGB color model. If the intensity of a particular channel is added to produce one color while it should be absent in the production of the other color, the two colors are not related in terms of the channel. For example, Red is added to produce yellow while Red should be absent in the production of black. Then yellow and black are not related in terms of the Red channel even though some yellow pixels and black pixels may share similar intensities in the Red channel. ***Second, the range of intensities of the channel is overlapped between the two colors***. To obtain the range of intensities for a particular color, we refer to the typical pixels of the color and observe the pixels collected for the study in this work. The resulting ranges can be found in our software. After applying the two-fold criterion, we can determine the color relations in terms of each channel, and hence three different graphs as in Table 3.1.

Note that gray is specially considered not to be related to any colors. Normally, gray would be related to the most of the colors because gray is produced by adding

the similar intensity in all the three channels, e.g., (128, 128, 128). However, the inclusion of the relations would make gray as a common neighbor for many pairs of colors that are not related, e.g., brown and orange in terms of the Red channel. To avoid bridging the difference between unrelated colors by gray, we make gray unrelated to any colors, which is demonstrated as a good design by our experiments.

3.4.4 Architecture Design

ColorMLP has the same kind of architecture as an MLP except for the GATs. For an MLP, we just need to specify the number of layers and the number of neurons in each layer. We design an MLP with one hidden layer because the 11 colors are not linearly separable in the raw data space. We do not include additional hidden layers in order to avoid overfitting. For the number of neurons in the hidden layer, we propose a number of a multiple of three because we will employ the concatenation of three matrices produced by three respective GATs as the weight matrix in the connection between the last two layers.

For the GATs, we customize their original design for our usage. Originally, GAT is a network with one hidden layer. Both the hidden layer and the output layer are implemented as a graph attention layer. A graph attention layer takes as input an adjacency matrix representing a graph and another matrix containing the features or representations of vertices, and produces a matrix of new representations of vertices. The new representation of each vertex is a weighted average of transformed representations of its adjacent vertices and itself, which is formally

realized as follows:

$$\boldsymbol{h}_i' = \sigma\left(\sum_{j \in N_i} \alpha_{ij} \boldsymbol{W} \boldsymbol{h}_j\right), \tag{3.1}$$

where $\boldsymbol{h}_i' \in \mathbb{R}^{D'}$ is the new representation of vertex i, σ is the sigmoid activation function, N_i is a set of vertices including vertices adjacent to vertex i and itself, $\alpha_{ij} \in \mathbb{R}$ named attention is a weight assigned to vertex j, $\boldsymbol{W} \in \mathbb{R}^{D' \times D}$ is a learnable matrix of parameters for transforming vertex representations, and $\boldsymbol{h}_j \in \mathbb{R}^D$ is the representation of vertex j in the input. To obtain the attention weight α_{ij}, an attention mechanism is designed. In particular, the attention weight is obtained by the following softmax function:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\boldsymbol{a}[\boldsymbol{W}\boldsymbol{h}_i || \boldsymbol{W}\boldsymbol{h}_j]\right)\right)}{\sum_{k \in N_i} \exp\left(\text{LeakyReLU}\left(\boldsymbol{a}[\boldsymbol{W}\boldsymbol{h}_i || \boldsymbol{W}\boldsymbol{h}_k]\right)\right)}, \tag{3.2}$$

where $\boldsymbol{a} \in \mathbb{R}^{2D'}$ is vector of a learnable parameters for computing attention coefficients, $||$ is the operation of concatenation, and LeakyReLU is an activation function [66]. Through stacking two graph attention layers, GAT was demonstrated to be the state-of-the-art model on several vertex classification tasks. One may refer to its paper [61] for more detailed explanations of the design.

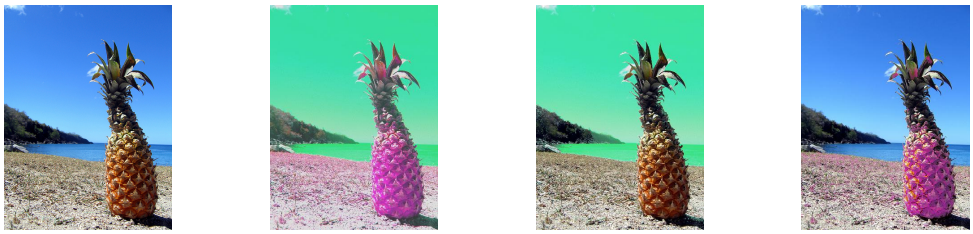
To incorporate GAT into ColorMLP, we mainly make two modifications. *First, we only utilize a single graph attention layer*. This is because in our case, we employ GAT only to learn the representations of the vertices, which is what the first graph attention layer of the original GAT does. *Second, we design a new kind of multi-head architecture*. The original multi-head architecture is designed as parallel multiple graph attention layers for learning vertex representations in order to stabilize the learning process. More specifically, the vertex representations

are the concatenation of multiple $\boldsymbol{h}_i'$ obtained in (3.1). In our design, we do not perform the concatenation of multiple heads, but perform the average of multiple heads. This is because the concatenation increases the dimensionality of the vertex representations, and we do not want to have a large dimensionality since the dimensionality corresponds to the number of neurons in the hidden layer of our MLP. Our experiments show that the proposed average strategy works well.

We then present the incorporation of the GATs into the MLP in a formal way. The $\boldsymbol{W}^r \in \mathbb{R}^{D' \times 11}$ in Fig. 3.2 contains the representations of 11 colors produced by the rGAT, which is a single-layered GAT with multiple heads averaged. Each column of $\boldsymbol{W}^r$ is obtained as follows:

$$\boldsymbol{W}_i^r = \frac{\sum_{k=1}^{K} \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k \boldsymbol{W}^k \boldsymbol{h}_j\right)}{K}, \tag{3.3}$$

where k and K are the index of a head and total number of heads, respectively. Similarly, $\boldsymbol{W}^g \in \mathbb{R}^{D' \times 11}$ and $\boldsymbol{W}^b \in \mathbb{R}^{D' \times 11}$ are produced by gGAT and bGAT, respectively. rGAT, gGAT and bGAT share the same architecture, and the only difference among them is in the input graph. As the name suggests, they take as input the three graphs corresponding to the three channels in Table 3.1, respectively. Note that there are no raw features of the 11 colors. We thus employ an identity matrix as the raw features, which is a common practice when applying graph neural networks to graphs without vertex features [67].



Original CJ PCJ(Sky) PCJ(Pineapple)

Figure 3.3: An example of an original image and images after the application of CJ or PCJ where PCJ(Sky) and PCJ(Pineapple) denote pixels of the sky and of the pineapple are selected, respectively.

3.5 Partial Color Jitter

This section presents the application of ColorMLP to data augmentation. In particular, color jitter (CJ) is a frequently studied data augmentation technique. CJ randomly changes the brightness, contrast, saturation and hue of an image, thereby increasing the diversity of the image. However, some studies [12], [13] show that CJ does not consistently improve the performance of image classification. One reason behind the failure of CJ is that some objects only have particular color patterns while CJ can make the color pattern arbitrary, resulting in objects that may never exist in the real world.

To reduce the risk of generating images with unrealistic colors, we propose partial color jitter (PCJ), which performs CJ only on a subset of pixels of an image. The subset of pixels is determined by randomly choosing a color and picking all the pixels whose label is classified as the color by ColorMLP. In this way, the aforementioned risk can be reduced by limiting the changes in color patterns to parts of an image. Moreover, the diversity brought by color changes is significantly increased because the changes made to an image are the joint results of CJ and the selection of a subset of pixels. To illustrate the difference between CJ and PCJ, we

Table 3.2: Number of pixels from different sources.

Dataset	Wikipedia	595C1	Hollasch	Survey	Total
red	30	33	14	15307	15384
green	68	147	20	51398	51633
blue	43	73	20	42736	42872
yellow	20	70	12	7795	7897
orange	36	32	7	9132	9207
brown	23	77	19	10504	10623
pink	48	0	0	12624	12672
purple	26	0	0	26412	26438
white	7	0	4	0	11
gray	13	63	2	0	78
black	5	0	1	1717	1723
All	319	495	99	177625	178538

present an example in Fig. 3.3. We can see that CJ may make the color patterns of the sky and the pineapple unrealistic, but PCJ can limit the unrealistic patterns to only the sky or only the pineapple. Besides, PCJ increases the diversity by selecting a subset of pixels.

3.6 Experiments on Computational Color Naming

In this section, we empirically evaluate the performance of the proposed ColorMLP on computational color naming.

3.6.1 Baselines, Implementations, and Datasets

Baselines. CCN is formulated as a classification problem in this study, and therefore can be solved by off-the-shelf classifiers, e.g., MLP and SVM. The objective of CCN is to discover a universal mapping from pixels into color names. The

nearest neighbor classifier employed by existing studies [49] on the fuzzy dataset of pixels can also be a baseline for discovering the universal mapping.

Implementations. We implement MLP and ColorMLP on PyTorch [68] framework, and directly utilize the SVM in scikit-learn. The detailed configurations are available in our source code.

Datasets. We collect pixel-color pairs from four sources, which are Wikipedia[2], 595C1[3], Hollasch [4] and Survey [5]. 595C1 is a U.S. federal standard for colors used in government procurement. Hollasch is popular dataset complied by a software developer Steve Hollasch. The Survey data was collected from a color survey in which over five million colors were named across 222,500 user sessions. The statistics of these datasets are summarized in Table 3.2. Note that these datasets contain pixels of color names other than the 11 names studied in this work. Moreover, there may exist other sources in the web. As a result, Table 3.2 just represents our best efforts of data collection for the current study.

We have two observations on the data. *First, the number of pixels in the Survey dataset is much larger than that in other datasets*. This is because the Survey dataset mainly contains pixels whose color names can be easily recognized by non-experts, e.g., (255, 255, 0) and (255, 0, 0) that are yellow and red, respectively. This kind of pixels usually lies on the surfaces of the RGB cube as illustrated in Fig. 3.1.

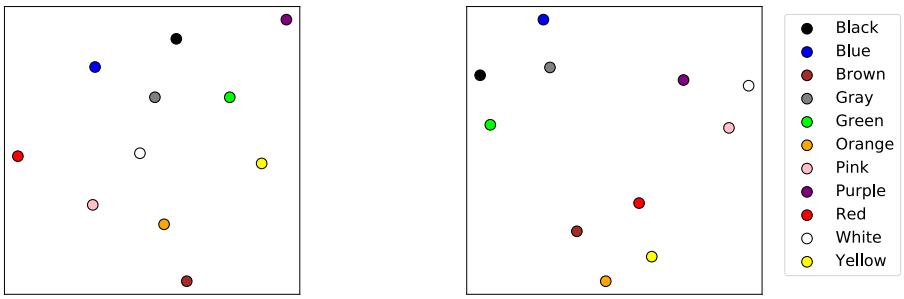
As a result, even though the Survey dataset is given by non-experts, it is highly likely that the names of the pixels are reliable. By contrast, other datasets contain

[2] https://en.wikipedia.org/wiki/Basic_Color_Terms
[3] https://people.csail.mit.edu/jaffer/Color/FED-STD-595C1.txt
[4] http://steve.hollasch.net/cgindex/color/colors.txt
[5] https://blog.xkcd.com/2010/05/03/color-survey-results/

pixels inside of the RGB cube whose names are given by experts. *Second, the data is class-imbalanced*. This is because the RGB color space itself is class-imbalanced. We can easily see that red pixels are many more than white pixels in the RGB cube. The class imbalance would lead to poor learning performance for the minority classes. An effective solution is to augment the data of the minority classes, and we adopt the SMOTE [69] algorithm. SMOTE basically generates synthetic data points along the line segment between two real data points. The way to select the two data points is based on a nearest neighbor based principle. Here, we make modifications to the way for selecting the real data points due to the uniqueness of our data. In particular, the majority of our data, about 99%, is from the Survey source, and the Survey source mainly contains pixels on the surface of the RGB cube. Therefore, the nearest neighbor based selection will result in synthetic pixels still lying on the surface of the cube, which may not be effective.

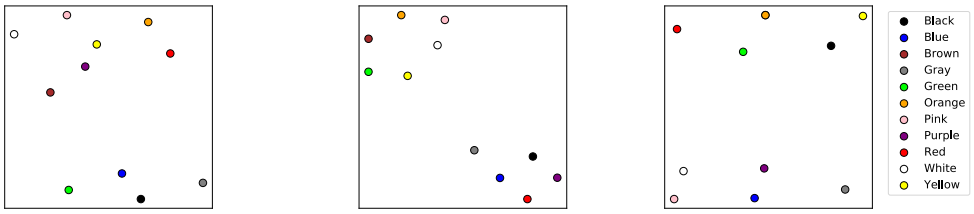
We modify the SMOTE algorithm as follows. For a particular minority class, we randomly select one pixel of the class from the Survey source and randomly select another pixel of the class from the rest of the sources. Then a synthetic pixel is randomly drawn along the line segment between the two pixels. In this way, synthetic pixels may not lie on the surface of the RGB cube because the pixels in the other sources lie inside of the RGB cube.

Note that there are other color spaces, e.g., HSL and CIELAB, and pixels can be converted among the spaces in a well-defined way. Here, we study pixels in the RGB space because we find that the RGB color model can be elegantly utilized as additional useful information by ColorMLP. Nevertheless, we have conducted experiments by using pixels in other color spaces, which show no significant differences from purely using pixels in the RGB space.



Visualization for MLP Visualization for ColorMLP

Figure 3.4: Visualizations of the color representations obtained from the MLP and ColorMLP in a two-dimensional space, respectively.



$\boldsymbol{W}^r$ $\boldsymbol{W}^g$ $\boldsymbol{W}^b$

Figure 3.5: Visualizations of the color representations $\boldsymbol{W}^r$, $\boldsymbol{W}^g$, $\boldsymbol{W}^b$ in a two-dimensional space, respectively.

3.6.2 Visualization of Color Representations

We first evaluate whether the color relations are preserved in ColorMLP or not. The color relations are embedded into color representations through the GATs. Each color representation is the concatenation of $\boldsymbol{W}_i^r$, $\boldsymbol{W}_i^g$ and $\boldsymbol{W}_i^b$. To demonstrate the effectiveness of our design, we use the t-SNE [70] tool to visualize the representations in a two-dimensional space as shown in Fig. 3.4. Besides, $\boldsymbol{W}_i^r$, $\boldsymbol{W}_i^g$ and $\boldsymbol{W}_i^b$ are separately visualized in Fig. 3.5. As a comparison, the visualization of each column of the corresponding weight matrix of the baseline MLP is also given. For this purpose, both ColorMLP and the MLP are trained on the entire set of pixels.

In the space, the smaller the distance between two points is, the more similar

Table 3.3: Classification accuracy of stratified five-fold cross validation from different models. The experiment with each model is repeated five times independently. Mean and standard deviation (Std.) are reported.

Model	Mean Accuracy $\pm$ Std. (%)
SVM	88.85 ± 0.03
MLP	90.37 ± 0.76
ColorMLP	90.37 ± 1.15

the two corresponding colors are. As we check the pair-wise distances, we can see that the visualization for ColorMLP is more meaningful than that for the MLP. For example, for the MLP, the distance between red and yellow is larger than that between red and green, and is even larger than that between red and blue. We know that the primary color Red is added to produce both red and yellow. Therefore, the distance between red and yellow should be the smallest among the three as shown in the visualization for colorMLP. For space consideration, we do not give more examples. As a conclusion, the color representations learned by colorMLP are effective in capturing the characteristics of the RGB color model.

3.6.3 Evaluation by Five-fold Cross Validation

We present the accuracy of stratified five-fold cross validation in Table 3.3. It shows ColorMLP and the MLP perform similarly, and both outperform the SVM, suggesting MLP is a better choice for the classification. Note that the Survey contributing about 99% of the data contains pixels mainly on the surfaces of RGB cube. Therefore, the evaluation by the five-fold cross validation on the pixels with color names can only provide limited information. In the following section, we design a better evaluation method.

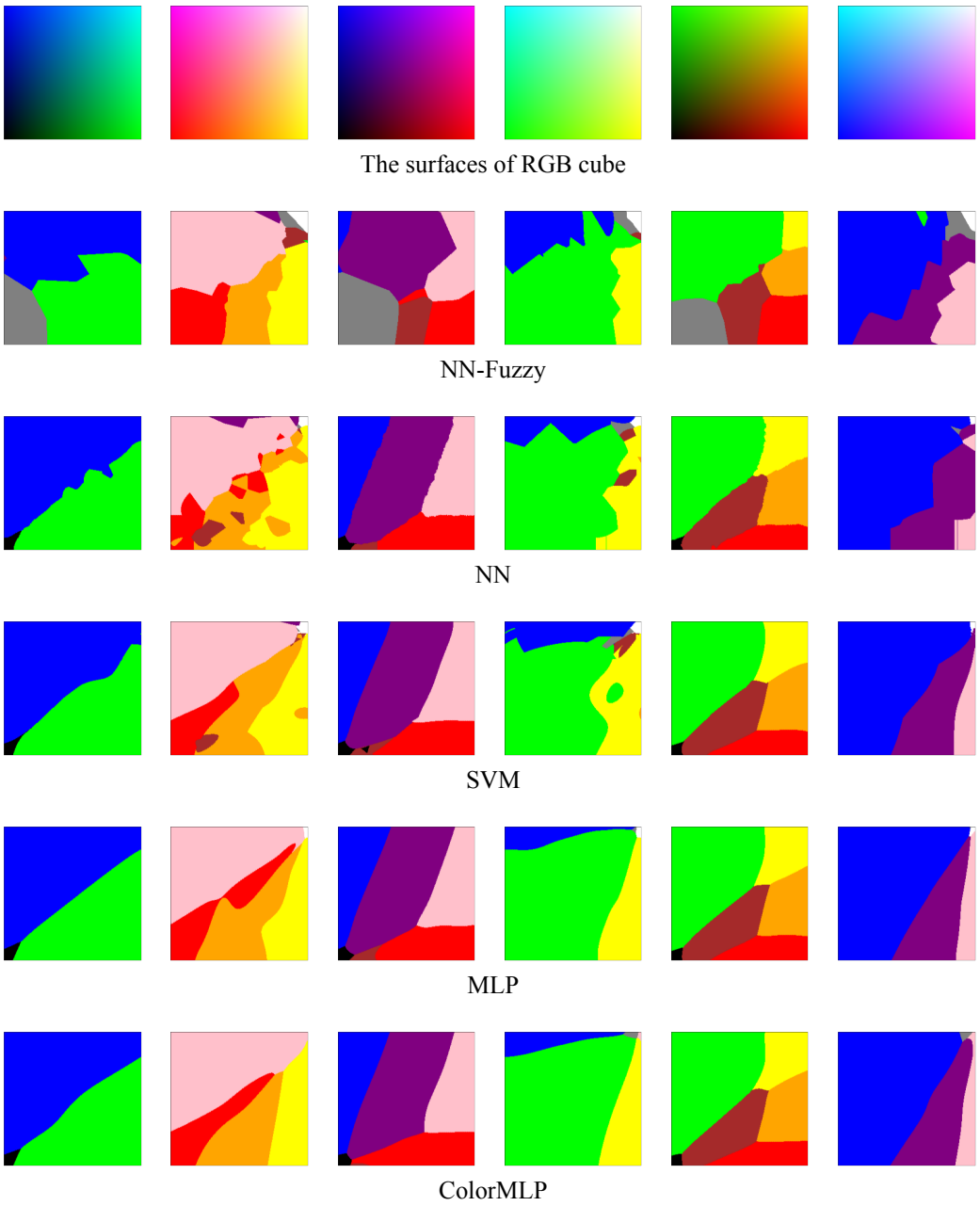


Figure 3.6: Visualization of color regions in the surface of the RGB cube. The first row consists of original appearance of the surfaces. All the other rows consist of color regions obtained by classification models where each region is comprised of only a representative pixel of the respective color. NN-Fuzzy is the approach of applying NN to the fuzzy dataset.

Figure 3.7: Visualization of color regions in the entire RGB cube.

3.6.4 Evaluation by Visualization

In this section, we perform the evaluation on the entire RGB cube by classifying all the pixels in the RGB cube. The classification is our objective of a universal color-name mapping. We first present the classification results for the pixels on the surfaces of the RGB cube in Fig. 3.6. In Fig. 3.6, we divide each surface into color regions where each region is comprised of only a representative pixel of the respective color for the purpose of evaluation.

NN-Fuzzy is the approach of applying the nearest neighbor (NN) classifier to a fuzzy dataset [48] and the NN classifier is a commonly used model for CCN with the fuzzy dataset [49]. In our setting, the color name of each pixel in the fuzzy dataset is chosen as the one with the largest membership value. NN is the approach of applying the NN classifier to our collected data. Note that NN is also usually employed by color-deficiency assistance Apps. The difference between the results obtained by NN-Fuzzy and NN mainly lies in whether there is a gray region. According to the RGB color model, gray pixels have similar intensities in all the three channels and therefore gray pixels should only exist in the diagonal within the RGB cube. Both NN-Fuzzy and NN obtain regions of very irregular shapes including disjoint components, which may be because the number of pixels with color names is quite small compared to the entire RGB space such that the named pixels sporadically distribute over the space. According to cognitive studies of colors [71], [72], the region of each color should have no disjoint components. All the models trained on the known pixel-color pairs perform better, and the proposed ColorMLP performs the best. In particular, the regions obtained by ColorMLP have no disjoint components and no strange protuberances. These dis-

joint components and strange protuberances of the baselines should be the results of overfitting.

Then we present the color regions in the entire RGB cube in Fig. 3.7. We can have similar observations as above. Note that the color regions actually consist of hundreds of thousands of pixels. Moreover, figuring out the boundary of the regions is the most challenging part of CCN, especially for these chromatic colors. As a result, we can conclude that ColorMLP significantly outperforms all the baselines.

3.7 Experiments on Image Classification

In this section, we evaluate the performance of the proposed PCJ on image classification.

3.7.1 Datasets, Baselines and Implementations

Datasets. We study three image classification datasets, which are *CIFAR-10*, *CIFAR-100* [73] and *ImageNet-ILSVRC2012* [74]. 1) *CIFAR-10* and *CIFAR-100* consist of colorful images in 32×32 resolution and are categorized into 10 and 100 classes, respectively. 2) *ImageNet-ILSVRC2012* is a large-scale dataset with 1000 classes and more than 1.2 million images at different resolutions.

Baselines. PCJ is compared to state-of-the-art(SoTA) color involved data augmentation methods. 1) *ISDA* [75] regularizes a deep model with implicit semantic data augmentation, e.g., changing the view angle and changing the object's color, by optimizing a novel loss function. 2) *ColorJitter* [12] randomly changes the color properties including the brightness, contrast, saturation and hue of the entire

Table 3.4: Top-1 test error rate (%) on CIFAR-10/CIFAR-100 dataset. Mean values and standard deviations are from four independent experiments. The best results are **bold-faced**.

Model	CIFAR-10				CIFAR-100			
	Baseline	ISDA	CJ	PCJ	Baseline	ISDA	CJ	PCJ
ResNet-20	7.98 ± 0.20	$\mathbf{7.78 \pm 0.16}$	9.06 ± 0.13	7.91 ± 0.12	30.62 ± 0.19	30.92 ± 0.18	33.00 ± 0.23	$\mathbf{30.12 \pm 0.19}$
ResNet-32	7.04 ± 0.24	7.01 ± 0.18	8.29 ± 0.13	$\mathbf{6.90 \pm 0.15}$	28.45 ± 0.47	28.67 ± 0.29	30.71 ± 0.22	$\mathbf{27.71 \pm 0.20}$
ResNet-44	6.36 ± 0.40	$\mathbf{5.89 \pm 0.41}$	7.20 ± 0.27	5.99 ± 0.26	25.44 ± 0.50	24.95 ± 0.43	27.65 ± 0.90	$\mathbf{24.57 \pm 0.20}$
ResNet-56	6.24 ± 0.34	6.04 ± 0.42	6.77 ± 0.39	$\mathbf{5.79 \pm 0.41}$	24.61 ± 0.32	24.54 ± 0.44	27.17 ± 0.56	$\mathbf{23.84 \pm 0.16}$
ResNet-110	5.80 ± 0.54	5.76 ± 0.46	6.64 ± 0.23	$\mathbf{5.41 \pm 0.30}$	$\mathbf{22.77 \pm 0.40}$	23.66 ± 0.23	26.71 ± 1.27	23.04 ± 0.10
ResNet-18-PreAct	5.53 ± 0.26	5.03 ± 0.09	6.19 ± 0.13	$\mathbf{4.99 \pm 0.04}$	23.97 ± 0.21	$\mathbf{23.32 \pm 0.48}$	26.17 ± 0.38	23.61 ± 0.20
DenseNet-BC-100-12	4.71 ± 0.22	4.85 ± 0.11	5.18 ± 0.15	$\mathbf{4.35 \pm 0.05}$	22.73 ± 0.44	21.79 ± 0.09	24.40 ± 0.23	$\mathbf{21.72 \pm 0.26}$
Wide-ResNet28-10	3.96 ± 0.10	3.74 ± 0.16	4.41 ± 0.14	$\mathbf{3.59 \pm 0.12}$	18.98 ± 0.26	$\mathbf{18.46 \pm 0.18}$	21.34 ± 0.17	18.48 ± 0.06

image. Besides, we also report the performances of different deep networks without any augmentation methods mentioned above as the *Baseline* shown in Table 3.4 and Table 3.5.

Implementations. For convolutional neural networks, we implement ResNet [76], DenseNet [77] and Wide-ResNet [78] on the CIFAR-10/100 dataset, and ResNet [76], DenseNet [77] on ImageNet-ILSVRC2012. Detailed training configurations for these networks are available in our source code. All the baselines follow the original implementations respectively. Specifically, we follow the implementation of CJ in [12], where we use the default value of 1.0 for the hyper-parameter s for all experiments. For a fair comparison, PCJ uses the same hyper-parameters as CJ.

3.7.2 Experiment Results

Main results. Table 3.4 reports the performance on the CIFAR-10 and CIFAR-100 datasets respectively. We follow the convention of existing papers by reporting the test error rate (the smaller the better). In general, compared with the baseline, CJ degrades the performance whereas PCJ consistently improve the performance across different deep networks. Besides, PCJ achieves the comparable

Table 3.5: Single crop error rates (%) on the validation set of ImageNet-ILSVRC2012. The better results are **bold-faced**.

Network	Top-1 / Top-5 Error Rate (%)	
	Baseline	PCJ
ResNet-50	23.30 / 6.85	**22.46 / 6.54**
ResNet-101	21.41 / 5.89	**21.05 / 5.62**
ResNet-152	21.24 / 5.74	**20.53 / 5.44**
DenseNet-BC-121	23.26 / 6.70	**22.97 / 6.54**

SoTA performance. Table 3.5 presents the performance of PCJ on the large scale ImageNet-ILSVRC2012 dataset. We can observe that PCJ consistently improves the generalization of different deep networks. For example, the top-1 error rate from ResNet-50 is reduced by 0.84% with PCJ, which is close to the performance of ResNet-101 without PCJ.

The degradation from CJ. Similar to existing studies [12], [13], we observe that CJ does degrade the performance of deep networks. As reported in Table 3.4, the performance of networks trained with CJ is worse than the baselines. The potential reason for this failure has been discussed earlier in Section 3.5, i.e., CJ arbitrarily changes the color patterns of the entire image, resulting in possibly misleading augmented samples.

The improvement from PCJ. Unlike CJ, PCJ can consistently bring improvement. The potential reason has been discussed in Section 3.5. In one sentence, PCJ increases the diversity of training images as well as has a low risk of generating misleading images.

Comparison with other methods. We also compare PCJ with the SoTA color-involved data augmentation method, ISDA. The results are reported in Table 3.4. We can observe that PCJ is comparable with the state-of-the-art performance.

Table 3.6: The Ablation Study for PCJ. The baseline is ResNet-18-PreAct. Top-1 test error rate (%) are reported. Mean values and standard deviations are from four independent experiments. The best results are **bold-faced**.

Setting	CIFAR-10	CIFAR-100
Baseline	5.53 ± 0.26	23.97 ± 0.21
PCJ(NN)	5.12 ± 0.07	24.01 ± 0.11
PCJ(SVM)	5.18 ± 0.12	23.94 ± 0.19
PCJ(MLP)	5.18 ± 0.19	23.69 ± 0.28
PCJ(ColorMLP)	$\mathbf{4.99 \pm 0.04}$	$\mathbf{23.61 \pm 0.20}$

Specifically, PCJ almost achieves the best results with some deep networks. It is worth mentioning that ISDA involves not only color transformations but also other transformations whereas PCJ only does color transformations. In summary, according to the results, we observe that PCJ is at least comparable with ISDA.

3.7.3 Ablation Study

To get a better understanding of the effectiveness of the color classification component in PCJ, we conduct an ablation study on the image classification. Specifically, we study PCJ with different color classifiers, i.e., 1) *PCJ(NN)*, 2) *PCJ(SVM)*, 3) *PCJ(MLP)* and 4) *PCJ(ColorMLP)*. Table 3.6 reports the results of ResNet-18-PreAct on the CIFAR-10 and CIFAR-100 datasets, respectively. We can see that almost all PCJs perform better than the baseline, and that *PCJ(ColorMLP)* achieves the best performance. This result further demonstrates the superiority of ColorMLP over other color classifiers.

Table 3.7: Averaged time cost of data loading for a single batch.

Method	Data Loading Time (ms)		
	CIFAR-10	CIFAR-100	ImageNet
Baseline	23.71	23.26	513.08
Baseline+CJ	34.53	33.82	632.80
Baseline+PCJ	37.42	36.34	637.32

3.7.4 Additional Time Cost

We show the additional time cost induced by PCJ is limited. In particular, Table 3.7 reports the averaged wall time of data loading for a single batch in our experiments. We can see that the data loading time of PCJ is very close to that of CJ.

3.8 Chapter Summary

In this work, we have formulated a novel classification problem for CCN, and have proposed ColorMLP to solve the problem. The problem for the first time incorporates the RGB color model that serves as domain knowledge about the production of colors. To utilize the domain knowledge, we construct three color graphs by following the RGB color model and design GATs to embed the three graphs into an MLP. We have conducted experiments to demonstrate the effectiveness of ColorMLP. We have further expanded the application of CCN to data augmentation by designing PCJ. Extensive experiments have shown that PCJ can consistently improve the performance of image classification.

The limitations mainly include two aspects. First, the Survey data may have unreliable pixel-color pairs due to the non-expert crowdsourcing. Second, ColorMLP is only designed for the 11 colors. Note that there can be many more color

names used in the real world. To make ColorMLP work for other color names, both the data and the color graphs need to be updated. Future work thus includes addressing these two limitations.

Chapter 4

Vision Loss Estimation using Fundus Photograph for High Myopia

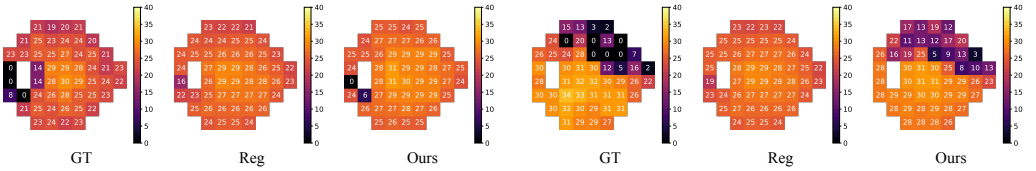
High myopia (HM) has become a global health issue as it causes various complications, such as myopic maculopathy (MM), resulting in irreversible vision loss. Visual field (VF) sensitivity is a commonly used metric to quantify vision loss, which is a crucial criterion for diagnosing HM-related complications. However, measuring VF is prohibitively time-consuming and subjective as it highly depends on patient compliance. Consequently, utilizing machine learning models to estimate VF becomes a feasible alternative. Fundus photographs have become the preferred modality for studying HM, due to their convenience of acquisition and incorporation of structural information. Conversely, estimating VF with vanilla regression using fundus photographs falls into trivial solutions. To tackle this challenge, we propose a novel method for VF estimation. In detail, we formu-

late VF estimation as an ordinal classification problem, where each VF point is interpreted as an ordinal variable rather than a continuous one, given that any VF point is a discrete integer with a relative ordering. Besides, we introduce an auxiliary task for MM classification to assist the generalization of VF estimation. MM is strongly associated with vision loss, and its symptoms can be observed from the fundus photographs directly; therefore, the model will be explicitly regularized by the auxiliary information if utilized properly. Not only does our method outperform vanilla baseline by 15% on a clinic-collected real-world dataset, but it can also be utilized to detect potential vision loss for HM cases in large-scale preliminary selection.

4.1 Introduction

Myopia has evolved into a global health issue that endangers vision. By 2050, 50% of the global population will be myopic, and 10% will have high myopic [79]. Various complications of high myopia (HM), such as cataract, glaucoma, retinal detachment, and myopic maculopathy (MM) can result in irreversible vision loss [80]. Vision loss is a crucial criterion for diagnosing these complications. Consequently, quantifying vision loss is essential so that prompt treatment can be administered.

Visual field (VF) sensitivity is a commonly used metric to quantify vision loss which can be measured by the visual field test. The visual field test is prohibitively time-consuming and subjective due to its high dependence on patient compliance [54]. During the automated static perimetry test, one type of visual field test, the subject (patient) is requested to press a button when he sees a light. Then the



(a) Moderate Vision Loss (b) Severe Vision Loss

Figure 4.1: Visualization of predictions from different methods. GT denotes the ground truth, Reg denotes the regression baseline, and Ours denotes our method.

machine will estimate his VF based on the lights to which he successfully and unsuccessfully responded. Typically, such progress takes longer than 10 minutes.

Consequently, utilizing machine learning models to estimate VF becomes a feasible alternative, because these models are capable of making fast predictions. To the best of our knowledge, existing approaches estimate VF only for the glaucomatous population by using different combinations of retinal thicknesses [54], [55]. Besides, some research works propose to utilize various types of fundus photographs to estimate quantitative measurements (e.g., MD value) in glaucoma [56], [57]. Notably, our studied population differs from theirs; ours is HM, which is more prevalent than glaucoma and demonstrates a global trend [79]. Besides, the pathological of HM is different from glaucoma, where HM is the local tear and glaucoma shows global degradation. Existing clinical research is mainly based on fundus photographs for HM [81], as they are convenient to obtain, typically take a few seconds to scan, and contain structural information such as retinal vascular, myopic macular, etc.

However, estimating VF with vanilla regression falls into trivial solutions, producing nonsense predictions. Specifically, we implement ResNet [76] to build a vanilla regression model to estimate VF using fundus photographs. We visualize its predictions on two representative examples, a large and a minor vision loss case.

As shown in Fig. 4.1, these predictions from vanilla regression exhibit a relatively simple and consistent pattern, regardless of the severity of the vision loss.

To tackle this challenge, we propose a novel method for estimating VF for HM using fundus photographs. In general, our method outperforms the vanilla regression by 15% on a clinic-collected real-world dataset, and it produces more meaningful predictions, as shown in Fig. 4.1. In detail, We first formulate VF prediction as an ordinal classification problem, where each VF point is interpreted as an ordinal variable rather than a continuous one, given that any VF point is a discrete integer with a relative ordering. Besides, due to the limited data, we introduce an auxiliary task for predicting the MM category to assist the generalization of VF estimation. MM is strongly associated with vision loss and its symptoms can be directly observed from fundus photographs [82]. By additionally considering the MM category, the model will be explicitly regularized by auxiliary information if appropriately utilized, thereby improving its performance.

Our contributions are summarized as follows:

- We propose a novel method for VF estimation based on fundus photographs for high myopia. To the best of our knowledge, our method is the first work to estimate VF using fundus photographs, which produces nontrivial solutions, i.e., successfully detects vision loss in estimated VF.
- We evaluate our method on a clinic-collected real-world dataset. And the experimental results demonstrate the effectiveness of our method, which outperforms the vanilla regression by 15%.
- Our method has a practical application in the clinic, as it can accurately estimate VF, therefore it can be utilized in large-scale preliminary selection

for potential vision loss of the HM population.

4.2 Literature Review

We discuss three types of related work w.r.t. to three topics, respectively, which are VF estimation, ordinal classification, and auxiliary learning.

In terms of VF estimation, existing works [54], [55] mainly utilize the retinal thickness as input data to estimate VF for the glaucomatous population. Besides, some research works [56], [57] utilize fundus photographs to estimate quantitative measurements (e.g., MD value) in glaucoma.

Ordinal classification (aka, rank learning) is utilized for predicting labels on the ordinal variable [83], [84]. Different from the category in classification, the ordinal variable contains ordinal information, i.e., there is a relative ordering among different scales. In this work, we formulate VF estimation as an ordinal classification problem, because each VF point follows the property of the ordinal variable.

Auxiliary learning is a special type of multi-task learning [7], [16]. Specifically, auxiliary learning typically introduces auxiliary tasks to help generalize the primary task. Besides, auxiliary learning only pays attention to the performance of the primary tasks, whereas multi-task learning aims to achieve comparable performance among all tasks [16]. Auxiliary learning has succeeded in many applications, including computer vision [7], [37], natural language processing [38], and speech recognition [39]. In this work, we introduce MM classification as the auxiliary task to help the generalization of VF estimation. Because existing clinic research [82] suggests that MM is strongly associated with vision loss and its symptoms can be directly observed from fundus photographs. Therefore, MM

is sufficiently related to vision loss.

4.3 Problem Formulation

Let $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{m}_i)\}$ denote the training set, where $\boldsymbol{x}_i \in \mathcal{X}$ denotes the fundus photography, $\boldsymbol{m}_i \in \mathcal{M}$ denotes its corresponded VF. And $\mathcal{A} = \{(\boldsymbol{x}_i, y_i)\}$ denotes the auxiliary set, where $y_i \in \mathcal{Y}$ denotes the MM category of a given $\boldsymbol{x}_i$. The objective is to learn a model $f : \mathcal{X} \rightarrow \mathcal{M}$ by utilizing both $\mathcal{D}$ and $\mathcal{A}$. The novelty of this formulation is additionally utilizing the auxiliary set to improve the model's generalization. And challenges mainly come from the following two aspects. First, how to design the model f, as mentioned earlier, vanilla regression falls into trivial solutions. Second, how to utilize the auxiliary set to assist the generalization of f, as the auxiliary information is not always helpful during the learning process, i.e., sometimes may interfere [17]–[19].

4.4 Proposed Method

In this section, we first present an overview of the proposed method. Then, we introduce the details of different components.

4.4.1 Overview

We present an overview of the proposed method in Fig. 4.2. Specifically, the primary task (denoted by $\mathcal{T}_{\text{pri}}$) is the VF estimation and the auxiliary task is MM classification (denoted by $\mathcal{T}_{\text{aux}}$). The proposed method is to solve $\mathcal{T}_{\text{pri}}$ with the assistance of $\mathcal{T}_{\text{aux}}$. We propose to parameterize the solution for $\mathcal{T}_{\text{pri}}$ and $\mathcal{T}_{\text{aux}}$ by two

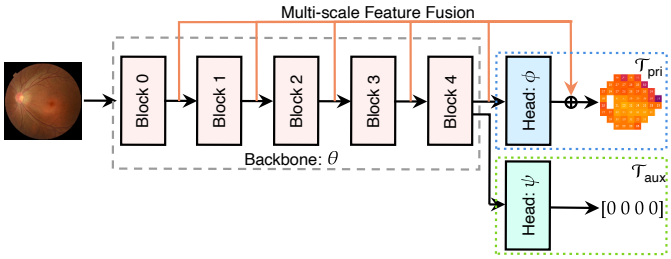


Figure 4.2: An overview of the proposed method. $\mathcal{T}_{\text{pri}}$ and $\mathcal{T}_{\text{aux}}$ denote the primary task and auxiliary task, respectively. And ϕ and ψ denote the task-specific parameters for $\mathcal{T}_{\text{pri}}$ and $\mathcal{T}_{\text{aux}}$, seperately. Both $\mathcal{T}_{\text{pri}}$ and $\mathcal{T}_{\text{aux}}$ share a same backbone parameterized by θ.

neural networks: $f(\cdot;\theta,\phi)$ and $g(\cdot;\theta,\psi)$, where they share the same backbone θ and have their own task-specific parameters ϕ and ψ. Following the auxiliary learning paradigm, the overall objective function is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{\text{pri}}(\theta,\phi) + \lambda\mathcal{L}_{\text{aux}}(\theta,\psi) \tag{4.1}$$

where $\mathcal{L}_{\text{pri}}$ and $\mathcal{L}_{\text{aux}}$ denote the loss function for $\mathcal{T}_{\text{pri}}$ and $\mathcal{T}_{\text{aux}}$, respectively. $\lambda \in (0,1]$ is a hyper-parameter to control the importance of $\mathcal{L}_{\text{aux}}$.

4.4.2 Primary Task: VFS Prediction

The overall interest is *only* the primary task $\mathcal{T}_{\text{pri}}$, which is to estimate the VF $\boldsymbol{m}_i$ using its fundus photograph $\boldsymbol{x}_i$. And $\mathcal{T}_{\text{pri}}$ is parameterized by $f(\cdot;\theta,\phi) : \mathcal{X} \to \mathcal{M}$. We observe that VF $\boldsymbol{m}_i$ has two distinct properties: 1) Discretization: $\forall m_i^j \in [0,40] \cap \mathbb{Z}$, that is, any VF value is a *positive discrete integer*. 2) Ordinalization: $m_i^0 \prec m_i^1 \prec ... \prec m_i^j$, there is a *relative order* among VF values. Therefore, we formulate $\mathcal{T}_{\text{pri}}$ as an *ordinal classification* (aka, *rank learning*) problem, where m_i^j represents an *ordinal variable/rank* rather than a continuous one. Fol-

lowing [83], [84], we extent the *ordinal variable/rank* into binary labels, i.e., $\boldsymbol{m}_i^j = [r_i^{j,1}, ..., r_i^{j,K-1}]$ where $r_i^{j,k} \in \{0, 1\}$ indicates whether $\boldsymbol{m}_i^j$ exceeds k-th rank or not. To achieve rank-monotonic and guarantee prediction consistency, we utilize the *ordinal bias* [84]. In detail, the task-specific parameter ϕ contains independent bias for each ordinal variable. Therefore, $\mathcal{T}_{\text{pri}}$ can be solved by the binary cross-entropy loss, which is defined as follows:

$$\mathcal{L}_{\text{pri}}(\theta, \phi) = \mathbb{E}_{(\boldsymbol{x}_i, \boldsymbol{m}_i) \in \mathcal{X} \times \mathcal{M}}[L_{\text{BCE}}(f(\boldsymbol{x}_i; \theta, \phi), \boldsymbol{m}_i)] \tag{4.2}$$

where $L_{\text{BCE}}(\cdot)$ denotes the binary cross-entropy loss

In addition, we propose to reuse the features from different blocks, as they contain distinct spatial information. Specifically, we propose Multi-scale Feature Fusion (MFF) for aggregating features from different blocks. As highlighted in orange in Fig. 4.2, MFF aggregates features from all blocks at the last in an addition operation. The detailed implementation is reported in Sec. 4.5.2.

4.4.3 Auxiliary Task: MM Classification

The auxiliary task $\mathcal{T}_{\text{aux}}$ is introduced *only* to assist the generalization of $\mathcal{T}_{\text{pri}}$. In detail, $\mathcal{T}_{\text{aux}}$ is to predict its MM catergory y_i based on fundus photograph $\boldsymbol{x}_i$, which is parameterized by $g(\cdot; \theta, \psi) : \mathcal{X} \to \mathcal{Y}$. MM is highly correlated to vision loss [82], therefore the model will be explicitly regularized if additionally utilizing the auxiliary information. And MM can be classified in order of increasing severity into five categories [85], i.e., $C_0 \prec C_1 ... \prec C_4$. Therefore, we also interpret the MM category as the *ordinal variable/rank*. Similar to the label extension in $\mathcal{T}_{\text{pri}}$, we extend the MM catergory into binary labels $\boldsymbol{y}_i = [r_1, r_2, r_3, r_4]$. The loss

function $\mathcal{L}_{\text{aux}}$ for solving $\mathcal{T}_{\text{aux}}$ is also the binary cross-entropy, which is defined as follows:

$$\mathcal{L}_{\text{aux}}(\theta, \phi) = \mathbb{E}_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{X} \times \mathcal{Y}}[L_{\text{BCE}}(g(\boldsymbol{x}_i; \theta, \psi), \boldsymbol{y}_i)] \tag{4.3}$$

However, the $\mathcal{T}_{\text{aux}}$ is not always helpful for $\mathcal{T}_{\text{pri}}$ because of the negative transfer [17]–[19]. The negative transfer refers to a problem that sometimes $\mathcal{T}_{\text{aux}}$ becomes harmful for $\mathcal{T}_{\text{pri}}$. Specifically, let $\nabla_\theta \mathcal{L}$ denote the gradient of Eq.(4.1) in terms of the shared parameters θ, and it can be decomposed as follows:

$$\nabla_\theta \mathcal{L} = \nabla_\theta \mathcal{L}_{\text{pri}} + \lambda \nabla_\theta \mathcal{L}_{\text{aux}} \tag{4.4}$$

$\mathcal{T}_{\text{aux}}$ becomes harmful for $\mathcal{T}_{\text{pri}}$, when the *cosine similarity* between $\nabla_\theta \mathcal{L}_{\text{pri}}$ and $\mathcal{L}_{\text{aux}}$ becomes negative [17], i.e., $\cos(\nabla_\theta \mathcal{L}_{\text{aux}}, \nabla_\theta \mathcal{L}_{\text{pri}}) < 0$. Negative transfer is observed in our setting when optimizing Eq.(4.1) directly, as illustrated in Fig. 4.3a.

Following [17], we block negative transfer by refining $\nabla_\theta \mathcal{L}_{\text{aux}}$. Specifically, we adapt the weighted cosine simiarily to refine $\nabla_\theta \mathcal{L}_{\text{aux}}$, which is defined as follows:

$$\nabla_\theta \mathcal{L}_{\text{aux}} := \max\left(0, \cos(\nabla_\theta \mathcal{L}_{\text{aux}}, \nabla_\theta \mathcal{L}_{\text{pri}})\right) \cdot \nabla_\theta \mathcal{L}_{\text{aux}} \tag{4.5}$$

4.5 Experiments

In this section, we conduct experiments on a clinic-collected real-world dataset to evaluate the performance of our proposed method.

4.5.1 The Studied Data

The studied data comes from a high myopia population, including 75 patients, each with diagnosis information for 2 eyes. For each eye, there are one fundus photograph and corresponding VF. Specifically, the fundus photograph is captured in colorful mode, and the VF is measured in the 24-2 mode (with 52 effective visual field sensitivity points). In addition, all fundus photographs have a label representing the MM category. Besides, 34 patients (i.e., 68 eyes) have SD-OCT scans in the macular region. For these eyes with SD-OCT scans, we extract the retinal thickness with the pre-trained model from [86]. According to whether the eye has SD-OCT scans or not, we divide the whole data into a training set and a test set. In detail, the training data and test data contain 68 eyes (from 34 patients) and 82 eyes (from 41 patients), respectively. It is worth mentioning that the training data and test data do not have the same patient. Besides, in the following K-fold cross-validation experiments, we split the training data based on the patient's ID to ensure that there is no information leakage, i.e., one eye of the same patient is in the training set and the other eye is in the validation set.

4.5.2 Experimental Setup

Data pre-processing. We choose the *left* eye pattern as our base. For fundus photographs, VFS and retinal thickness are not in *left* eye pattern, we convert them into the *left* eye pattern using the horizontal flip.

Data augmentation. Following [87], we consolidate a set of data augmentations for both fundus photographs and retinal thickness, respectively. The details are reported in Table.4.1 and Table.4.2, separately. Different from applying *all*

Table 4.1: Data augmentation for fundus photographs.

	Augmentation	Description
A	No transformation	Only normalize original fundus image to $[0., 1.]$
B	Rotation	Randomly rotate fundus photographs in $[-15°, 15°]$.
C	Shift	Randomly translate horizontally and vertically by up to 10% of the fundus image's height and width.
D	Scale	Randomly scale sampled from the interval $[0.9, 1.1]$.
E	Brightness, Contrast and Saturation	Modify the brightness, contrast and saturation by a random factor $[0.75, 1.25]$.
F	All transformations	Apply transformations from A to E
G	TrivialAugment	Sample an augmentation from A to E uniformly at random and applies it with its own strength, which is sampled uniformly at its own range.

Table 4.2: Data augmentation for retinal thickness.

	Augmentation	Description
A	No transformation	Only normalize original fundus image to $[0., 1.]$
B	Rotation	Randomly rotate fundus image in $[-15°, 15°]$.
C	Shift	Randomly translate horizontally and vertically by up to 10% of the fundus image's height and width.
D	Scale	Randomly scale sampled from the interval $[0.9, 1.1]$.
E	All transformations	Apply transformations from A to E
F	TrivialAugment	Sample an augmentation from A to F uniformly at random and applies it with its own strength, which is sampled uniformly at its own range.

[87] augmentations during training, we utilize the *TrivialAugment* [25] instead. *TrivialAugment* randomly selects one from the given data augmentations, which generates more diverse augmented data.

Evaluation methods. For quantitative evaluation, we utilize two metrics: *root mean squared error* (RMSE) and *mean absolute error* (MAE), following [54]–[56], [88], [89]. For qualitative evaluation, we visualize two representative predictions on the test set, including a moderate case and a severe case, according to

the degree of visual loss. More visualized results are presented in the supplementary material.

Baseline methods. We mainly compare our approach to the vanilla regression model. Specifically, we first compare our approach to the vanilla regression model using fundus photographs. Besides, we compare our approach to the vanilla regression model using different retinal thicknesses. In detail, we consider two thickness variants following existing works [54], [88]: (a) the combination of the thickness of ganglion cell and inner plexiform layer (GCIPL), retinal nerve fiber layer (RNFL) and rod and cone layer (RCL) [88]. (b) the combination of GCIPL and RNFL [54]. (c) only the thickness of RNFL [55]. Due to the limited data, we compare our method with vanilla regression using different retinal thicknesses on K-fold cross-validation on training data.

Implementation details. We utilize the ResNet-18 [76] as the baseline model. For the vanilla regression baseline, we use only one *linear* layer at last. For our proposed classification baseline, we use the combination of *Conv2D*, *BatchNorm2D* and *ReLU* as the classification head for $\mathcal{T}_{\text{pri}}$. For the multi-scale feature fusion, we utilize the above classification head to reuse features from different blocks, then aggregate all transformed features at last in addition operation. Note that the features from earlier blocks have relatively large features, and we use *AdaptiveAvgPooling2D* to perform downsampling before feeding into the classification head to reduce the computational parameters. For $\mathcal{T}_{\text{aux}}$, we use only one *linear* layer as the classifier. For a fair comparison, we train all methods with the same training configurations. Specifically, we train the models with 80 training epochs and the SGD optimizer, where the learning rate is set to 0.01, momentum is set to 0.9 and L2 weight decay is set to $1e^{-4}$. Besides, we utilize a cosine learning

Table 4.3: Main results. 'K-fold' indicates performance from K-fold cross-validation on training data, where we split the training data into K fold based on the patient's ID to ensure no data leakage. 'Test' indicates performance on test data (training on training data). (↓) denotes the lower value indicates better performance. And the better results are **bold-faced**.

Method	Modality	K-fold(K=5)		Test	
		RMSE (↓)	MAE (↓)	RMSE (↓)	MAE (↓)
Regression	Thickness-(a)	4.94 ± 0.23	3.12 ± 0.05	-	-
Regression	Thickness-(b)	4.80 ± 0.17	3.04 ± 0.12	-	-
Regression	Thickness-(b)	4.86 ± 0.22	3.13 ± 0.18	-	-
Regression	Fundus	4.62 ± 0.07	2.95 ± 0.07	4.28 ± 0.03	2.89 ± 0.06
Ours(λ=0.1)	Fundus	**4.44 ± 0.27**	**2.78 ± 0.10**	**3.69 ± 0.03**	**2.41 ± 0.04**

rate decay [90] to adjust the learning rate per epoch. Finally, we fix all input resolutions to 384×384 for both training and evaluation. All experiments are run independently with four seeds: 0, 1, 2, and 3. As for hyper-parameters, we search them on training data with K-fold cross-validation.

4.5.3 Experimental Results

Main results. Table 4.3 reports the performance of our methods and different baselines. In general, compared to baselines, our approach achieves the best performance. Compared to the baseline using fundus, our method outperforms it by 13.79% and 16.61% according to the RMSE and MAE on test data. Besides, our method achieves better performance than vanilla regression models using different retinal thicknesses, as demonstrated by the performance on K-fold cross-validation on training data. In addition, we observe that the regression baseline using fundus photographs achieves better performance than those using different retinal thicknesses, which follows the argument from [81].

Visualization of predictions. As shown in Fig. 4.1, we visualize two rep-

Table 4.4: Ablation study on main components. OR denotes the ordinal classification baseline. MFF denotes multi-scale feature fusion. AUX denotes the auxiliary task. BNT denotes blocking negative transfer from Eq.(4.5).

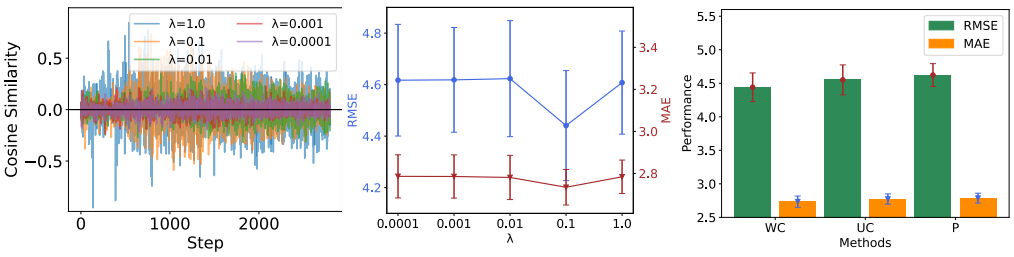
OR	MFF	AUX	BNT	RMSE(↓)	MAE(↓)
✓	✓	✓	✓	**3.69 ± 0.03**	**2.41 ± 0.04**
✓	✓	✓		3.74 ± 0.02	2.46 ± 0.03
✓	✓			3.73 ± 0.04	2.45 ± 0.02
✓				3.77 ± 0.02	2.49 ± 0.03

resentative cases from the test data, one with moderate vision loss and one with severe vision loss. In general, we find that our method outperforms the vanilla regression baseline in terms of estimating vision loss, whereas vanilla regression fails. In particular, the predictions from the vanilla regression baseline share a simple and trivial pattern for both cases. Both predictions appear to be very similar, but neither predicts actual vision loss. In contrast, our method estimates more precisely, as its predictions are accurate to the ground truth, revealing the actual vision loss.

4.5.4 Ablation Study

To get a better understanding of the effectiveness of the main components in our proposed method, we conduct a series of ablation studies.

Effectiveness of main components. We first examine the effectiveness of the main components by ablating them. The results are reported in Table 4.4. In general, we can observe that all components can improve performance. Specifically, MFF aggregates the distinct spacial information from multi-scale features, thereby improving the models' performance. Besides, simply adding auxiliary tasks brings a degradation, because of the existence of negative transfer. Meanwhile, with the



(a) Negative transfer (b) Impact of λ (c) Blocking negative transfer

Figure 4.3: Visualization of (a) Negative transfer when optimizing Eq.(4.1) directly, (b) Impact of hyper-parameter λ, and (c) Different methods for blocking the negative transfer.

help of blocking negative transfer by gradient refinement from Eq.(4.5), the improvements from the auxiliary task can be significantly improved.

Impact of hyper-parameter λ. We study the impact of the hyper-parameter λ with K-fold cross validation on training data. We choose $\lambda \in \{1.0, 0.1, 0.01, 0.001, 0.0001\}$. According to the results shown in Fig. 4.3b, we observe that $\lambda = 0.1$ achieves the best performance because at this time RMSE and MAE are the lowest.

Different methods for blocking the negative transfer. We also study different methods for blocking the negative transfer from the auxiliary task. We consider three alternative criteria for refining the auxiliary gradient: (1) weighted cosine (WC) similarity [17] (2) unweighted cosine (UC) similarity [17] (3) projection (P) [18]. In general, both of them are utilized to quantify whether the negative transfer from the auxiliary task exists or not. In detail, both (1) WC and (UC) modify the gradients from the auxiliary task by referring to the cosine similarity between gradients from the primary and auxiliary tasks. And (3) P projects gradients from the auxiliary task to the primary task, then removes the gradients whose direction is different from the primary task. For a fair comparison, we set $\lambda = 0.1$, then

conduct experiments on training data with K-fold cross-validation. The results are shown in Fig. 4.3c. We observe that (1) WC achieves the best performance among these methods.

4.6 Chapter Summary

In this work, we propose a novel method for estimating VF based on fundus photographs, which achieves superior performance and produces more meaningful predictions than the vanilla baseline. Besides, our method has a practical application in the clinic, that is, it can be utilized in large-scale preliminary selection for potential vision loss of the HM population. The major limitations of our method are two aspects. First, we utilize each eye from a patient as unique input, which neglects the similarity of the eyes, as they come from the same patient. Besides, we only utilize the MM category in the auxiliary task. Based on these limitations, further work can be improved in the following directions. First, exploring regularization for modeling the relationship between two eyes from the same patient is one of the future directions, which can further improve the model's generalization. Second, discovering more helpful auxiliary labels for the auxiliary tasks could be a new future direction. In addition, self-supervised auxiliary learning may be a more promising direction, because it performs auxiliary learning in a self-supervised manner, eliminating the need for manually discovering auxiliary labels.

Chapter 5

Conclusions, Open Challenges and Future Directions

5.1 Conclusion

In this thesis, We study auxiliary supervision for regularizing deep learning-based image classification. We first review the background and motivations, then We investigate the challenges. The challenges mainly lie in two regularizations, including data augmentation and auxiliary learning. Besides, We review existing works on data augmentation and auxiliary learning. In addition, We review the applications for computational color naming and vision loss estimation.

In Chapter 3, We present the first work on computational color naming (CCN) and further expand CCN's application to data augment. In detail, We propose a novel model named ColorMLP for CCN by additionally utilizing the RGB Color Model as regularization. Besides, We expand CCN's application to data augmentation by designing a color jittering-based data augmentation method, namely Partial

Color Jitter, which performs CJ on a subset of pixels belonging to the same color of an image. In this way, PCJ partially changes the color properties of images, thereby significantly increasing images' diversity. At last, We conduct experiments to show that PCJ has a remarkable regularization effect on image classification tasks.

In Chapter 4, We present the second work on vision loss estimation. We first review the problem in vision loss estimation and find out that existing vanilla baselines produce trivial solutions and thus fail to estimate vision loss accurately. To tackle this challenge, we propose a novel method based on the characteristics of Visual field (VF) sensitivity data. In order to achieve better performance, We introduce an auxiliary task for myopic maculopathy classification to assist the generalization of vision loss estimation. Finally, we conduct experiments to evaluate our method on a clinic-collected real-world dataset.

5.2 Open Challenges

There are several open challenges remaining to be addressed in terms of data augmentation and auxiliary learning.

First, theoretical research on data augmentation is one open challenge. Existing theoretical works [91], [92] are limited to label-preserving data augmentation, i.e., data augmentation does not change the label of data. For these label-covarying data augmentations, such as Mixup [8] and its variant [35], there is no such theoretical framework to get deep insight and explore their benefit and functionality.

In addition, analyzing the effect of data augmentation on the transferability of pre-trained DNNs is an open challenge. Pre-trained DNNs become the preferred

initialization for many downstream tasks; however, data augmentation typically introduces priors or biases, so whether it improves or decreases the transferability of pre-trained DNNs remains a question.

Besides, designing a data augmentation mainly relies on priors, however, there are no efficient ways to determine whether the priors are adequate or not. Existing works mainly assume the selected priors are beneficial and helpful, then conduct experiments to verify them, which is inevitably time-consuming. Besides, data augmentations are usually closely connected to the data/modality's intrinsic characteristic; therefore design data augmentation also heavily relay on task-specific domain knowledge, hence its generality is not usually guaranteed.

As for auxiliary learning, finding sufficient related auxiliary tasks is a basic challenge. Auxiliary tasks are typically determined manually based on domain knowledge or assumptions, and validating them is exceptionally costly. Both improving the efficiency of validating auxiliary tasks and automatically finding auxiliary tasks are open challenges.

In addition, auxiliary tasks are not always guaranteed to have a positive impact on the primary task, as existing works[17]–[19] have found their negative transfer. Therefore, how to eliminate the negative transfer from auxiliary tasks becomes an open challenge.

5.3 Future Directions

Based on the above open challenges, there are some potential future directions that are valuable for exploration.

First, from the representation learning perspective to analyze data augmenta-

tion is one interesting future direction. Data augmentation mainly transforms or augments the data in input space, which is a high-dimensional space. Data in high dimensional space typically contains redundant information, whereas its features in latent space, a relatively low dimensional space, remain compact. Therefore, analyzing data augmentation from the representation learning perspective in latent space is more appropriate.

In addition, based on the representation learning perspective, latent space data augmentation is a promising future direction. Input space data augmentation augments data over the input space, where domain knowledge and priors usually need to be validated first. Besides, input space data augmentation naturally couples with data modality, whereas latent space data augmentation is modality-agnostic; more specifically, latent space data augmentation generates or augments deep features in the latent space.

In terms of auxiliary learning, self-supervised auxiliary learning is a promising future direction to explore. The basic challenge in auxiliary learning is to first find sufficient related auxiliary tasks and then verify them, which requires domain knowledge or assumptions and is costly. If auxiliary learning is performed in a self-supervised manner, where the auxiliary labels are obtained from the data itself, then the above manual procedure is no longer needed, hence it will be more efficient and more general.

Chapter 6

References

[1] L. Nanni, S. Ghidoni, and S. Brahnam, "Handcrafted vs. non-handcrafted features for computer vision classification," *Pattern Recognition*, vol. 71, pp. 158–172, 2017.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[4] V. Shankar, R. Roelofs, H. Mania, A. Fang, B. Recht, and L. Schmidt, "Evaluating machine accuracy on imagenet," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 8634–8644.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[6] J. Kukačka, V. Golkov, and D. Cremers, "Regularization for deep learning: A taxonomy," *arXiv preprint arXiv:1710.10686*, 2017.

[7] S. Liu, A. J. Davison, and E. Johns, "Self-supervised generalisation with meta auxiliary learning," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019, pp. 1677–1687.

[8] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018.

[9] R. Wightman, H. Touvron, and H. Jégou, "Resnet strikes back: An improved training procedure in timm," *arXiv preprint arXiv:2110.00476*, 2021.

[10] M. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Sixth Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008*, IEEE Computer Society, 2008, pp. 722–729.

[11] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, *Caltech-ucsd birds-200-2011 (cub-200-2011)*, `http://www.vision.caltech.edu/datasets/cub_200_2011/`, 2011. (visited on 01/05/2023).

[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 1597–1607.

[13] S. Mounsaveng, I. Laradji, I. B. Ayed, D. Vazquez, and M. Pedersoli, "Learning data augmentation with online bilevel optimization for image classification," in *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, IEEE, 2021, pp. 1690–1699.

[14] T. Xiao, X. Wang, A. A. Efros, and T. Darrell, "What should not be contrastive in contrastive learning," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[15] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019.

[16] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[17] Y. Du, W. M. Czarnecki, S. M. Jayakumar, M. Farajtabar, R. Pascanu, and B. Lakshminarayanan, "Adapting auxiliary losses using gradient similarity," *arXiv preprint arXiv:1812.02224*, 2018.

[18] Vivien, *Learning through auxiliary tasks*, https://vivien000.github.io/blog/journal/learning-though-auxiliary_tasks.html, Feb. 17, 2019. (visited on 01/05/2023).

[19] L. M. Dery, Y. N. Dauphin, and D. Grangier, "Auxiliary task update decomposition: The good, the bad and the neutral," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[20] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[21] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.

[22] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, 2019, pp. 113–123.

[23] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast autoaugment," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, vol. 32, 2019, pp. 6662–6672.

[24] D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel, "Population based augmentation: Efficient learning of augmentation policy schedules," in *Proceedings of the 36th International Conference on Machine Learning, ICML*

2019, 9-15 June 2019, Long Beach, California, USA, ser. Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 2731–2741.

[25] S. G. Müller and F. Hutter, "Trivialaugment: Tuning-free yet state-of-the-art data augmentation," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, IEEE, 2021, pp. 754–762.

[26] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[27] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, AAAI Press, 2020, pp. 13 001–13 008.

[28] P. Chen, S. Liu, H. Zhao, and J. Jia, "Gridmask data augmentation," *arXiv preprint arXiv:2001.04086*, 2020.

[29] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, IEEE, 2019, pp. 6022–6031.

[30] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," in *8th International Conference on Learning Repre-*

sentations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.

[31] Y. Wang, X. Pan, S. Song, H. Zhang, G. Huang, and C. Wu, "Implicit semantic data augmentation for deep networks," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019, pp. 12 614–12 623.

[32] M. Li, F. Su, O. Wu, and J. Zhang, "Logit perturbation," in *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, AAAI Press, 2022, pp. 1359–1366.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[34] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using GAN for improved liver lesion classification," in *15th IEEE International Symposium on Biomedical Imaging, ISBI 2018, Washington, DC, USA, April 4-7, 2018*, IEEE, 2018, pp. 289–293.

[35] V. Verma, A. Lamb, C. Beckham, *et al.*, "Manifold mixup: Better representations by interpolating hidden states," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 6438–6447.

[36] T. Cheung and D. Yeung, "MODALS: modality-agnostic automated data augmentation in the latent space," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[37] T. Mordan, N. Thome, G. Hénaff, and M. Cord, "Revisiting multi-task learning with ROCK: A deep residual auxiliary block for visual detection," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018, pp. 1317–1329.

[38] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 2013, pp. 3111–3119.

[39] S. Toshniwal, H. Tang, L. Lu, and K. Livescu, "Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, ISCA, 2017, pp. 3532–3536.

[40] M. Jaderberg, V. Mnih, W. M. Czarnecki, *et al.*, "Reinforcement learning with unsupervised auxiliary tasks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[41] Y. Burda, H. Edwards, D. Pathak, A. J. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019.

[42] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "Region-based image retrieval with high-level semantic color names," in *11th International Conference on Multi Media Modeling (MMM 2005), 12-14 January 2005, Melbourne, Australia*, IEEE Computer Society, 2005, pp. 180–187.

[43] F. S. Khan, R. M. Anwer, J. Van De Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez, "Color attributes for object detection," in *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, IEEE Computer Society, 2012, pp. 3306–3313.

[44] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, IEEE Computer Society, 2014, pp. 1090–1097.

[45] F. S. Khan, R. M. Anwer, J. van de Weijer, M. Felsberg, and J. Laaksonen, "Compact color-texture description for texture classification," *Pattern Recognition Letters*, vol. 51, pp. 16–22, 2015.

[46] J. M. G. Lammens, "A computational model of color perception and color naming," Ph.D. dissertation, State University of New York at Buffalo, 1994.

[47] A. Mojsilovic, "A computational model for color naming and describing color composition of images," *IEEE Transactions on Image Processing*, vol. 14, no. 5, pp. 690–699, 2005.

[48] R. Benavente, M. Vanrell, and R. Baldrich, "A data set for fuzzy colour naming," *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, vol. 31, no. 1, pp. 48–56, 2006.

[49] R. Benavente, M. Vanrell, and R. Baldrich, "Parametric fuzzy sets for automatic color naming," *Journal of the Optical Society of America A*, vol. 25, no. 10, pp. 2582–2593, 2008.

[50] G. Menegaz, A. Le Troter, J. Sequeira, and J.-M. Boi, "A discrete model for color naming," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–10, 2006.

[51] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512–1523, 2009.

[52] L. Yu, Y. Cheng, and J. van de Weijer, "Weakly supervised domain-specific color naming based on attention," in *24th International Conference on Pattern Recognition, ICPR 2018, Beijing, China, August 20-24, 2018*, IEEE Computer Society, 2018, pp. 3019–3024.

[53] Z. Yuan, B. Chen, J. Xue, N. Zheng, *et al.*, "Illumination robust color naming via label propagation," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, IEEE Computer Society, 2015, pp. 621–629.

[54] K. Park, J. Kim, and J. Lee, "A deep learning approach to predict visual field using optical coherence tomography," *PLOS ONE*, vol. 15, no. 7, pp. 1–19, 2020.

[55] S. Datta, E. B. Mariottoni, D. Dov, A. A. Jammal, L. Carin, and F. A. Medeiros, "RetiNerveNet: Using recursive deep learning to estimate pointwise 24-2 visual field data based on retinal structure," *Scientific Reports*, vol. 11, no. 1, pp. 1–10, 2021.

[56] M. Christopher, C. Bowd, A. Belghith, *et al.*, "Deep learning approaches predict glaucomatous visual field damage from oct optic nerve head en face images and retinal nerve fiber layer thickness maps," *Ophthalmology*, vol. 127, no. 3, pp. 346–356, 2020.

[57] J. Lee, Y. W. Kim, A. Ha, *et al.*, "Estimating visual field loss from monoscopic optic disc photography using deep learning model," *Scientific reports*, vol. 10, no. 1, pp. 1–10, 2020.

[58] B. Berlin and P. Kay, *Basic color terms: Their universality and evolution*. Univ of California Press, 1991.

[59] J. Birch, "Worldwide prevalence of red-green color deficiency," *JOSA A*, vol. 29, no. 3, pp. 313–320, 2012.

[60] R. Hirsch, *Exploring colour photography: a complete guide*. Laurence King, 2005.

[61] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018.

[62] R. Khan, J. Van de Weijer, F. Shahbaz Khan, D. Muselet, C. Ducottet, and C. Barat, "Discriminative color descriptors," in *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, IEEE, 2013, pp. 2866–2873.

[63] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, 2019, pp. 5177–5186.

[64] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin, "Learning semantic-specific graph representation for multi-label image recognition," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, IEEE, 2019, pp. 522–531.

[65] V. Kurková, "Kolmogorov's theorem and multilayer neural networks," *Neural Networks*, vol. 5, no. 3, pp. 501–506, 1992.

[66] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the International Conference on Machine Learning, ICML 2013*, ser. Proceedings of Machine Learning Research, PMLR, vol. 30, 2013, p. 3.

[67] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[68] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Process-*

ing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 8024–8035.

[69] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[70] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[71] D. Roberson, J. Davidoff, I. R. Davies, and L. R. Shapiro, "Color categories: Evidence for the cultural relativity hypothesis," *Cognitive Psychology*, vol. 50, no. 4, pp. 378–411, 2005.

[72] T. Regier, P. Kay, and N. Khetarpal, "Color naming reflects optimal partitions of color space," *Proceedings of the National Academy of Sciences*, vol. 104, no. 4, pp. 1436–1441, 2007.

[73] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[74] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, IEEE Computer Society, 2009, pp. 248–255.

[75] Y. Wang, G. Huang, S. Song, X. Pan, Y. Xia, and C. Wu, "Regularizing deep networks with semantic data augmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[76] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, pp. 770–778.

[77] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, IEEE Computer Society, 2017, pp. 2261–2269.

[78] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*, BMVA Press, 2016.

[79] B. A. Holden, T. R. Fricke, D. A. Wilson, *et al.*, "Global prevalence of myopia and high myopia and temporal trends from 2000 through 2050," *Ophthalmology*, vol. 123, no. 5, pp. 1036–1042, 2016.

[80] T. Y. Wong, A. Ferreira, R. Hughes, G. Carter, and P. Mitchell, "Epidemiology and disease burden of pathologic myopia and myopic choroidal neovascularization: An evidence-based systematic review," *American Journal of Ophthalmology*, vol. 157, no. 1, 9–25.e12, 2014.

[81] D. C. Hood and C. G. De Moraes, "Challenges to the common clinical paradigm for diagnosis of glaucomatous damage with oct and visual fields," *Investigative Ophthalmology & Visual Science*, vol. 59, no. 2, pp. 788–791, 2018.

[82] R. Silva, "Myopic maculopathy: A review," *Ophthalmologica*, vol. 228, no. 4, pp. 197–213, 2012.

[83] L. Li and H. Lin, "Ordinal regression by extended binary classification," in *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, MIT Press, 2006, pp. 865–872.

[84] W. Cao, V. Mirjalili, and S. Raschka, "Rank consistent ordinal regression for neural networks with application to age estimation," *Pattern Recognition Letters*, vol. 140, pp. 325–331, 2020.

[85] O. Xiao, X. Guo, D. Wang, *et al.*, "Distribution and severity of myopic maculopathy among highly myopic eyes," *Investigative ophthalmology & visual science*, vol. 59, no. 12, pp. 4880–4885, 2018.

[86] A. G. Roy, S. Conjeti, S. P. K. Karri, *et al.*, "Relaynet: Retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks," *Biomedical Optics Express*, vol. 8, no. 8, pp. 3627–3642, 2017.

[87] D. Bar-David, L. Bar-David, S. Soudry, and A. Fischer, "Impact of data augmentation on retinal oct image segmentation for diabetic macular edema analysis," in *Ophthalmic Medical Image Analysis - 8th International Workshop, OMIA 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings*, ser. Lecture Notes in Computer Science, vol. 12970, Springer, 2021, pp. 148–158.

[88] Y. Zheng, L. Xu, T. Kiwaki, *et al.*, "Glaucoma progression prediction using retinal thickness via latent space linear regression," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &*

Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019, ACM, 2019, pp. 2278–2286.

[89] L. Xu, R. Asaoka, T. Kiwaki, H. Murata, Y. Fujino, and K. Yamanishi, "Pami: A computational module for joint estimation and progression prediction of glaucoma," in *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, ACM, 2021, pp. 3826–3834.

[90] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, 2019, pp. 558–567.

[91] S. Chen, E. Dobriban, and J. H. Lee, "A group-theoretic framework for data augmentation," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[92] T. Dao, A. Gu, A. Ratner, V. Smith, C. D. Sa, and C. Ré, "A kernel theory of modern data augmentation," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 1528–1537.